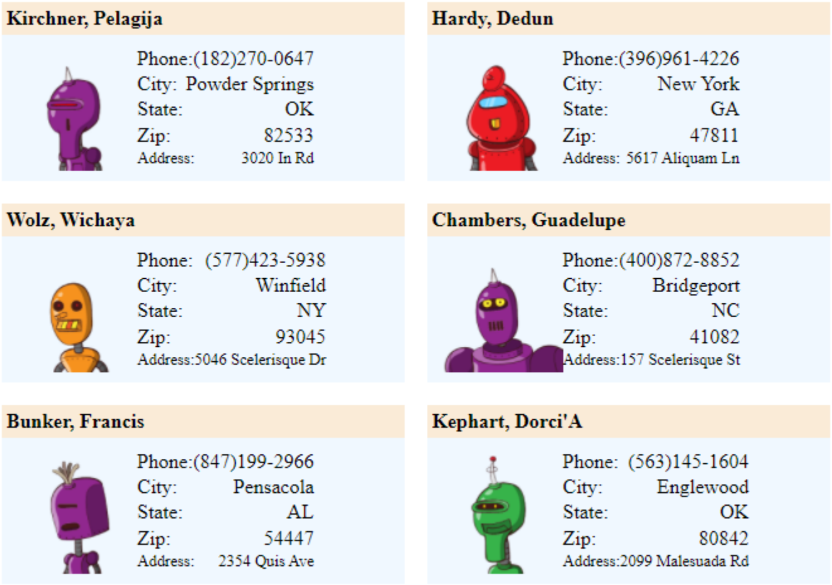
Get with AJAX

Get data from servers

Create a single project ajax-proj with few pages, each contains some information coming via AJAX

1. **address-book.html - Use Vanilla XMLHttpRequest**



Kirchner, Pelagija
Phone: (182)270-0647
City: Powder Springs
State: OK
Zip: 82533
Address: 3020 In Rd

Hardy, Dedun
Phone: (396)961-4226
City: New York
State: GA
Zip: 47811
Address: 5617 Aliquam Ln

Wolz, Wichaya
Phone: (577)423-5938
City: Winfield
State: NY
Zip: 93045
Address: 5046 Scelerisque Dr

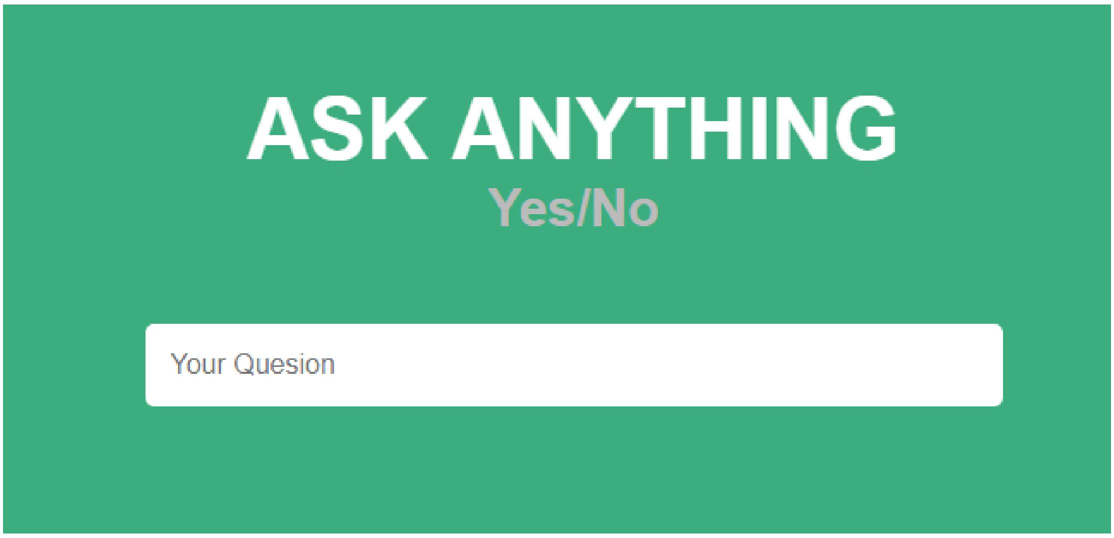
Chambers, Guadelupe
Phone: (400)872-8852
City: Bridgeport
State: NC
Zip: 41082
Address: 157 Scelerisque St

Bunker, Francis
Phone: (847)199-2966
City: Pensacola
State: AL
Zip: 54447
Address: 2354 Quis Ave

Kephart, Dorci'A
Phone: (563)145-1604
City: Englewood
State: OK
Zip: 80842
Address: 2099 Malesuada Rd

1. The FillText API (http://www.filltext.com/) is a nice option for having a server returning stuff for testing purpose.
2. Use the following data sample
3. Use it to get a **list of contacts** and show them on your page
4. For the images, use: https://robohash.org/
   a. Note – do not use AJAX for getting the image, just point the image src to a URL like: `https://robohash.org/`**`{someName}`**`?set=set5`

2. yes-no.html – Use jquery



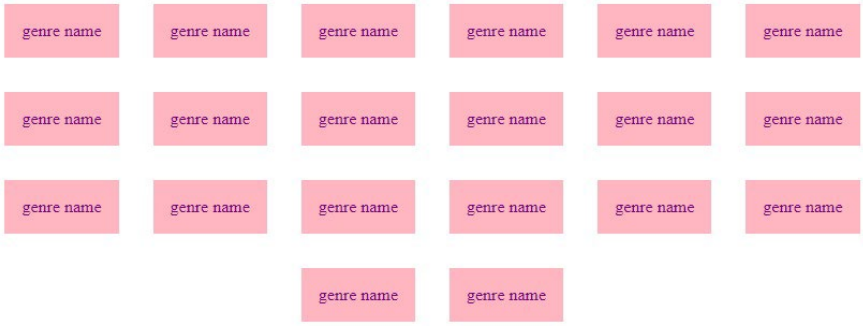
ASK ANYTHING

Yes/No

Your Quesion

- let the user ask a Yes/No question
- When the question has at least 3 characters and ends with a question mark, issue an AJAX call.
- Use the Yes/No API to get an answer
- After getting a response, render the response and the image to the user and also – issue another AJAX call:
    - If answer was YES – show a joke from this API
    - If answer was NO – give him a random dog from this API

3. movies.html – Use Vanilla XMLHttpRequest



genre name	genre name	genre name	genre name	genre name	genre name
genre name	genre name	genre name	genre name	genre name	genre name
genre name	genre name	genre name	genre name	genre name	genre name
		genre name	genre name		

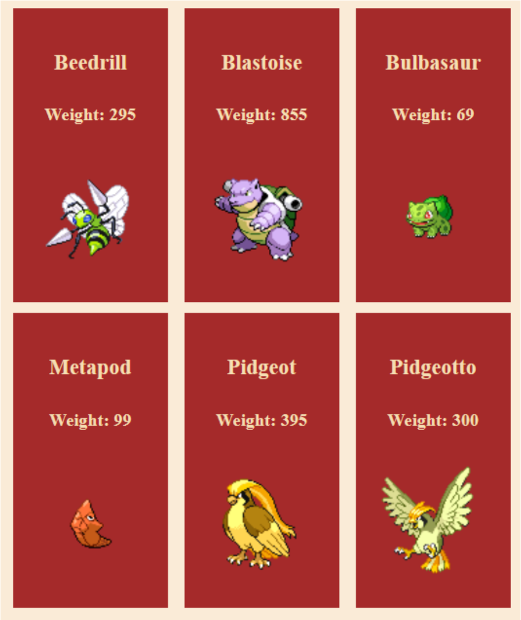
- Create an account here: https://www.themoviedb.org/signup
- Then, create an API key here: https://www.themoviedb.org/settings/api

- Show a list of genres https://api.themoviedb.org/3/genre/movie/list?api_key=API_KEY (Just the Name)
- when a genre is selected, issue another request and show the following details about the corresponding movies: Title, Overview, image.
  The call will look something like this https://api.themoviedb.org/3/discover/movie?api_key=API_KEY&with_genres=GENRE_ID
  NOTE: In order to display the images, you have to follow the instructions here: https://developers.themoviedb.org/3/getting-started/images
- Important: avoid reaching the calls-limit of this API by keeping some data in localStorage as a caching mechanism (the API will block if getting too many calls)
- Add a Reload button that calls the getGenres function again for re-displaying the data

4. **pokemons.html API – use jQuery**
   Show a list of Pokemons from this pokemon API

   - Start by showing sorted Pokemons' names
   - To show the image and weight, we will need to issue an AJAX call per Pokemon

   1. BONUS – use all the sprite images with a setInterval to create an img-by-img animation.
   2. BONUS – Allow the user to export a CSV of the pokemons, containing the name, weight and image/images of the pokemons



Beedrill

Weight: 295

Blastoise

Weight: 855

Bulbasaur

Weight: 69

Metapod

Weight: 99

Pidgeot

Weight: 395

Pidgeotto

Weight: 300

Done?!

Watch this