





service-worker



Version-manager



service-worker

- Version number.
- List of ressources to be cached.
- Installation logic
- Caching logic
- Fetch interception for cached items



version-manager



service-worker



version-manager

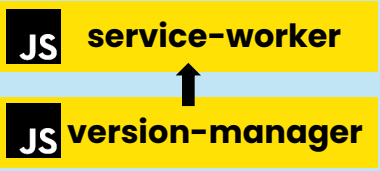
- Environnement check (web or PWA?)
- Captures the beforeinstallprompt event
- Triggers the installation logic through user input
- Registers/updates the service worker



service-worker



version-manager





JS service-worker




JS version-manager

User is presented with Index



Index

 PWAkedex [Home](#) [Pokemon](#) [Games](#) [Items](#)


Pokémon in cache: 1350/1350 [Install PWA?](#)

Welcome to PWAkedex


pokemon search

Advanced Search


Search




Bulbasaur
ID: 1
Grass Poison




Ivysaur
ID: 2
Grass Poison




Venusaur
ID: 3
Grass Poison




Charmander
ID: 4
Fire




Charmeleon
ID: 5
Fire



Charizard
ID: 6
Fire Flying



Squirtle
ID: 7
Water



Wartortle
ID: 8
Water



JS service-worker



JS version-manager

User is asked if they want to install.
If denied, a cookie prevents further asking



Index

The screenshot shows the PWAdex website interface. At the top, there's a navigation bar with the PWAdex logo and links for Home, Pokémon, Games, and Items. A status bar on the right indicates 'Pokémon in cache: 1350/1350' and an 'Install PWA?' button. A modal dialog titled 'Install as PWA?' is displayed in the center, asking for confirmation to install the app for offline use, with 'Install' and 'Close' buttons. Below the modal is a search bar labeled 'pokemon search' with an 'Advanced Search' dropdown and a 'Search' button. The main content area features a grid of eight Pokémon cards, each with a pixel art illustration, name, ID number, and type(s).

Image	Name	ID	Type(s)
	Bulbasaur	ID: 1	Grass, Poison
	Ivysaur	ID: 2	Grass, Poison
	Venusaur	ID: 3	Grass, Poison
	Charmander	ID: 4	Fire
	Charmeleon	ID: 5	Fire
	Charizard	ID: 6	Fire, Flying
	Squirtle	ID: 7	Water
	Wartortle	ID: 8	Water



JS service-worker

JS version-manager

If accepted, version-manager installs the PWA



Index

Install as PWA?



Installing the app as a PWA allows for offline use

Install

Close

Name or Number

Search



JS service-worker


JS version-manager



If accepted, version-manager installs the PWA



Index

 PWAkedex [Home](#) [Pokemon](#) [Games](#) [Items](#)





















Pokémon in cache: 1350/1350 [Install PWA?](#)

Welcome to PWAkedex

pokemon search

Advanced Search

Search

 <div>Bulbasaur ID: 1  Grass  Poison</div>	 <div>Ivysaur ID: 2  Grass  Poison</div>	 <div>Venusaur ID: 3  Grass  Poison</div>	 <div>Charmander ID: 4  Fire</div>
 <div>Charmeleon ID: 5  Fire</div>	 <div>Charizard ID: 6  Fire  Flying</div>	 <div>Squirtle ID: 7  Water</div>	 <div>Wartortle ID: 8  Water</div>



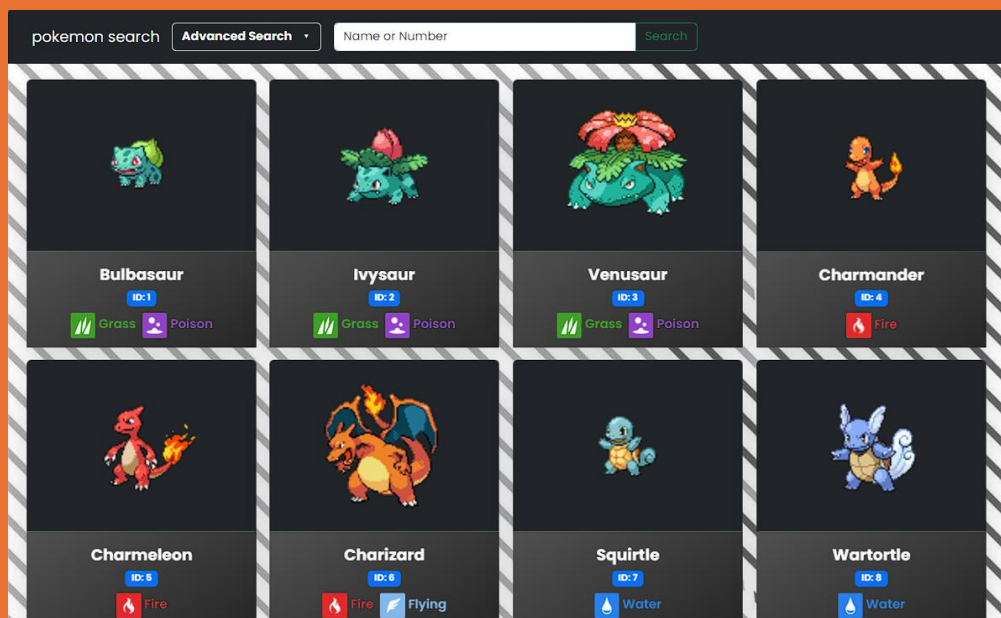
JS service-worker

JS version-manager

The grid of pokemon is rendered by grid-render



Index



JS

grid-render



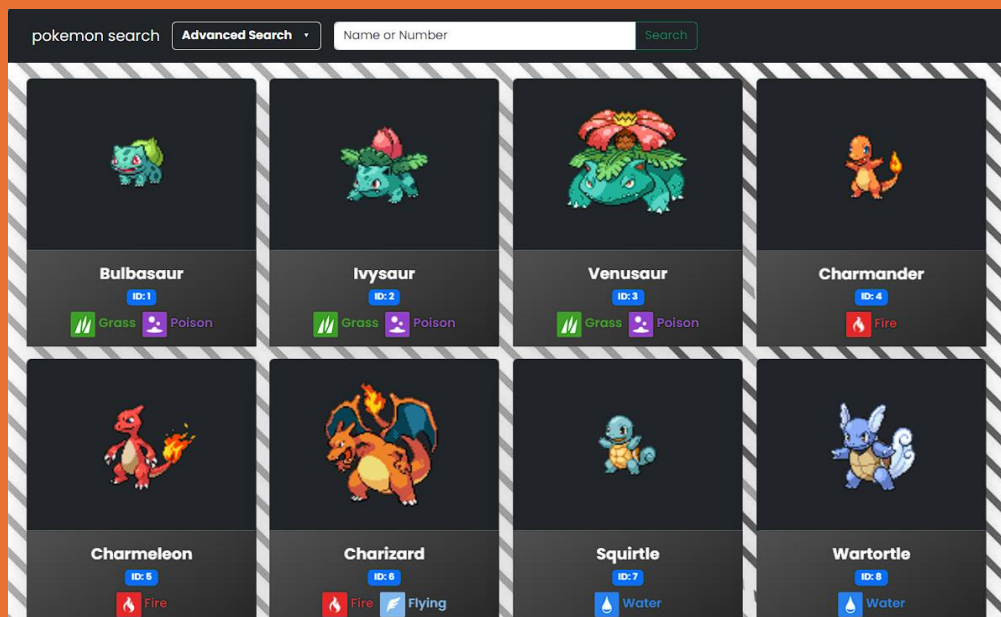
JS service-worker

JS version-manager

On page load, grid-render is asked to render 8 cards*



Index

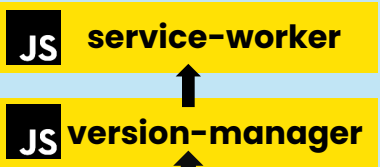


Asks for 8 pokemon

JS

grid-render

*Technically grid-render tells itself, but this is more visually pleasing



Index

pokemon search

Advanced Search

Search

Bulbasaur
ID: 1
Grass Poison

Ivysaur
ID: 2
Grass Poison

Venusaur
ID: 3
Grass Poison

Charmander
ID: 4
Fire

Charmeleon
ID: 5
Fire

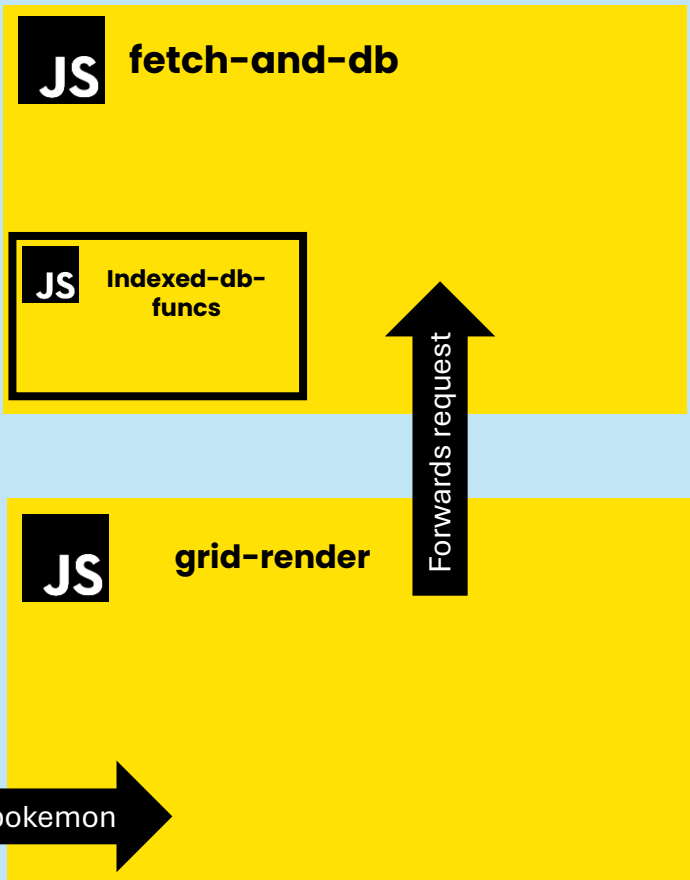
Charizard
ID: 6
Fire Flying

Squirtle
ID: 7
Water

Wartortle
ID: 8
Water

Asks for 8 pokemon

grid-render needs information to render to the card,
it asks fetch-and-db to provide information on a
Pokémon by name or id





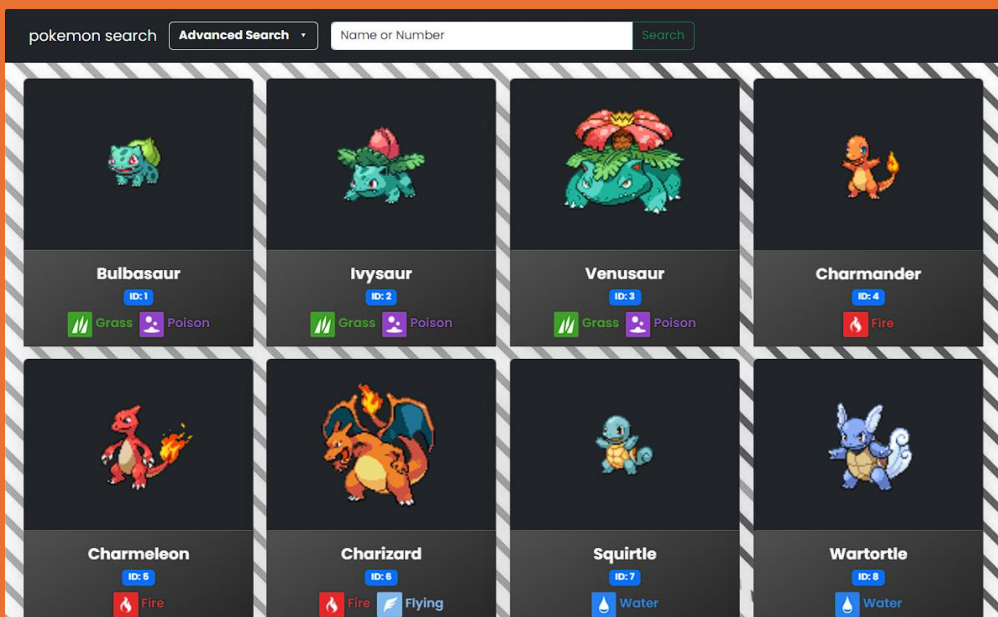
JS service-worker

JS version-manager

Data can come from the API or cache



Index



Asks for 8 pokemon

JS fetch-and-db

JS Indexed-db-funcs

Forwards request

JS grid-render



Dexie.js

IndexedDB



Dexie.js internal API handles Indexed DB requests for cleaner code



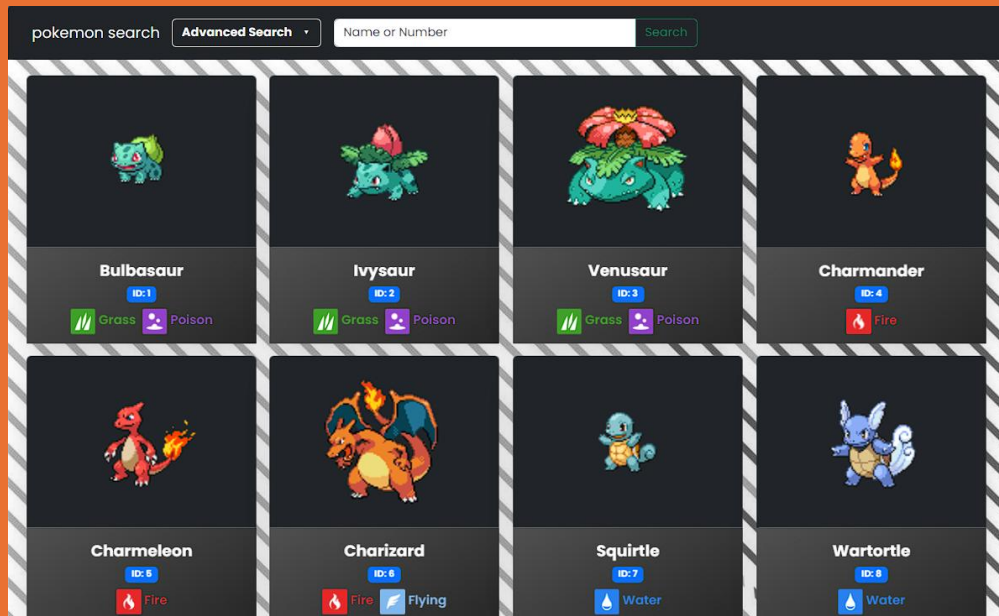
JS service-worker

JS version-manager

First, we check if we have the json in cache



Index



Asks for 8 pokemon

JS fetch-and-db

Checks cache

JS Indexed-db-funcs

Forwards request

JS grid-render



Dexie.js

IndexedDB



Dexie.js internal API handles Indexed DB requests for cleaner code

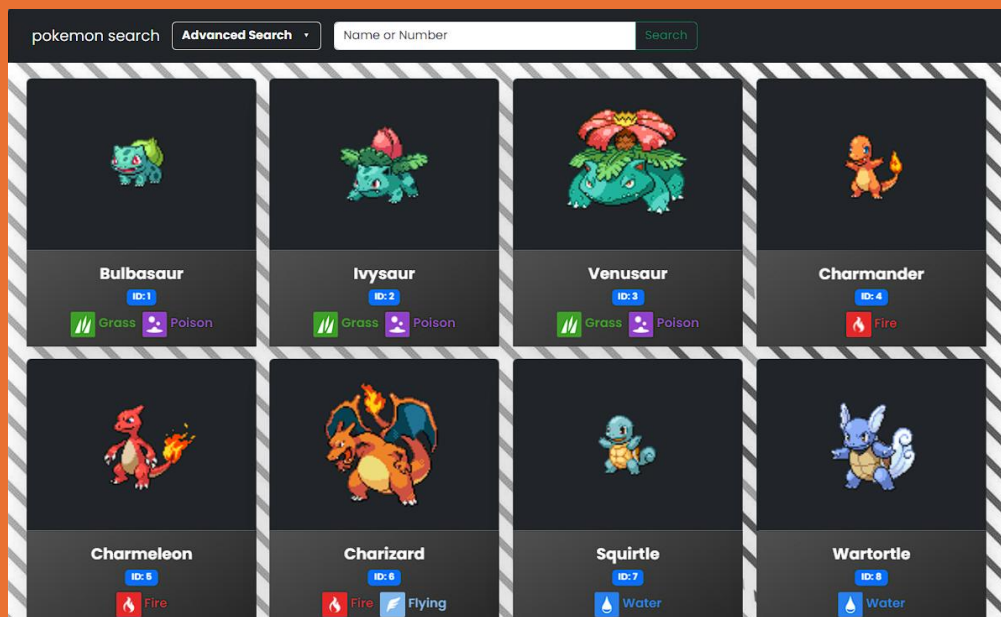


JS service-worker

JS version-manager



Index



Asks for 8 pokemon

JS fetch-and-db

JS Indexed-db-funcs

JS grid-render

Forwards request

Checks cache

NOT FOUND

Dexie.js

IndexedDB



Dexie.js internal API handles Indexed DB requests for cleaner code

PokeAPI



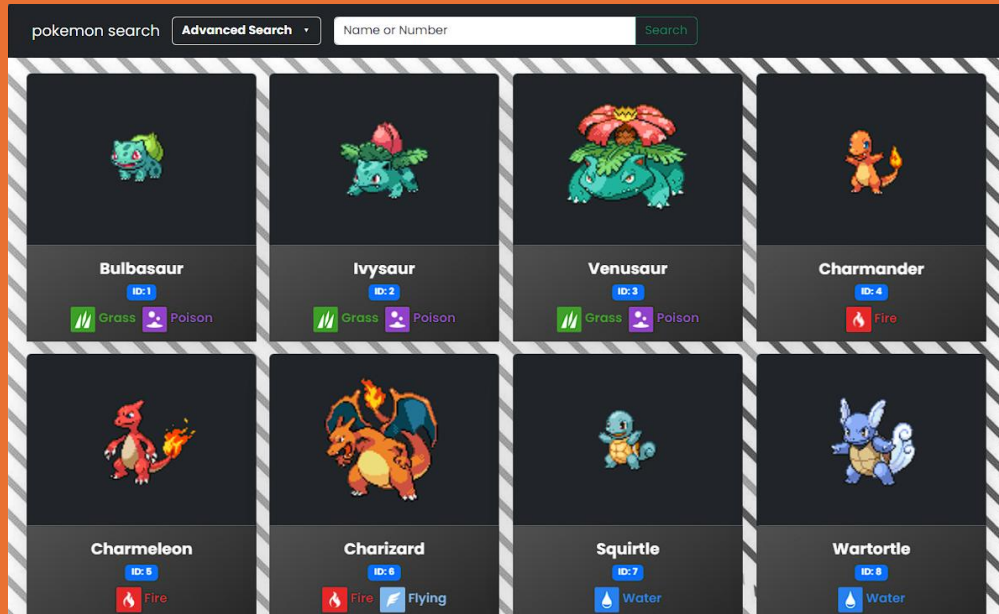
JS service-worker

JS version-manager



Index

If not, we ask the API



Asks for 8 pokemon

JS fetch-and-db

Fetch from API

JS Indexed-db-funcs

Forwards request

JS grid-render

PokeAPI

Dexie.js

IndexedDB





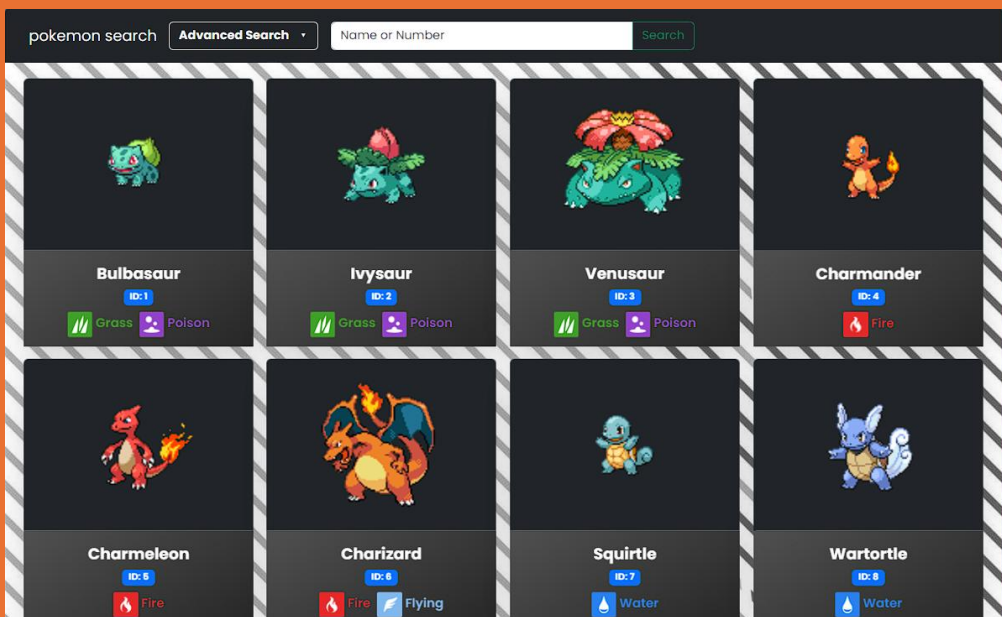
JS service-worker

JS version-manager

Data is passed back to fetch-and-db



Index



Asks for 8 pokemon

JS Fetch-and-DB

Fetch from API

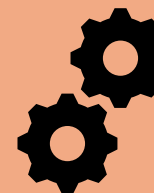
Response

JS Indexed-db-funcs

Forwards request

JS grid-render

PokeAPI



Dexie.js

IndexedDB





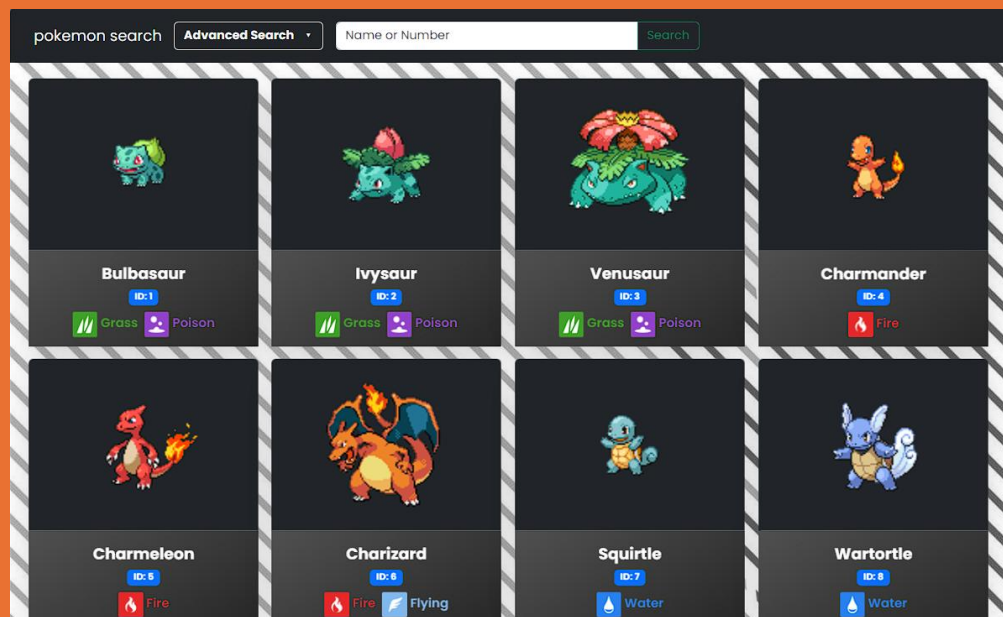
JS service-worker

JS version-manager

fetch-and-db saves the json to indexedDB



Index



Asks for 8 pokemon

JS fetch-and-db

Fetch from API

Response

JS Indexed-db-funcs

Save response to DB

JS grid-render

Forwards request

PokeAPI

Dexie.js

IndexedDB





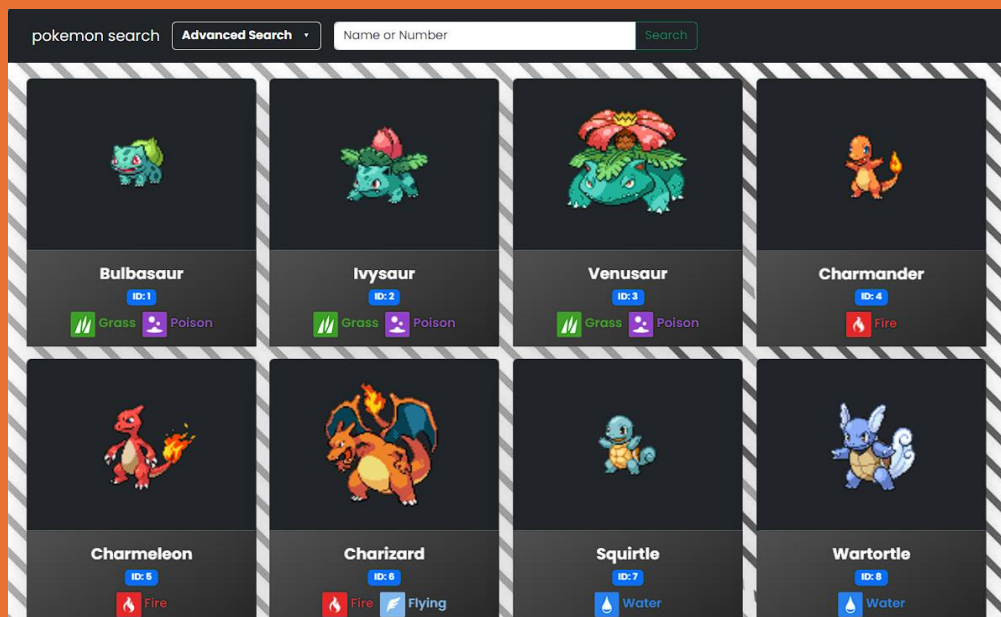
JS service-worker

JS version-manager

Data is passed back to grid-render



Index



Asks for 8 pokemon

JS fetch-and-db

Fetch from API

Response

JS Indexed-db-funcs

Save response to DB

JS grid-render

Forwards request

Return data

PokeAPI

Dexie.js

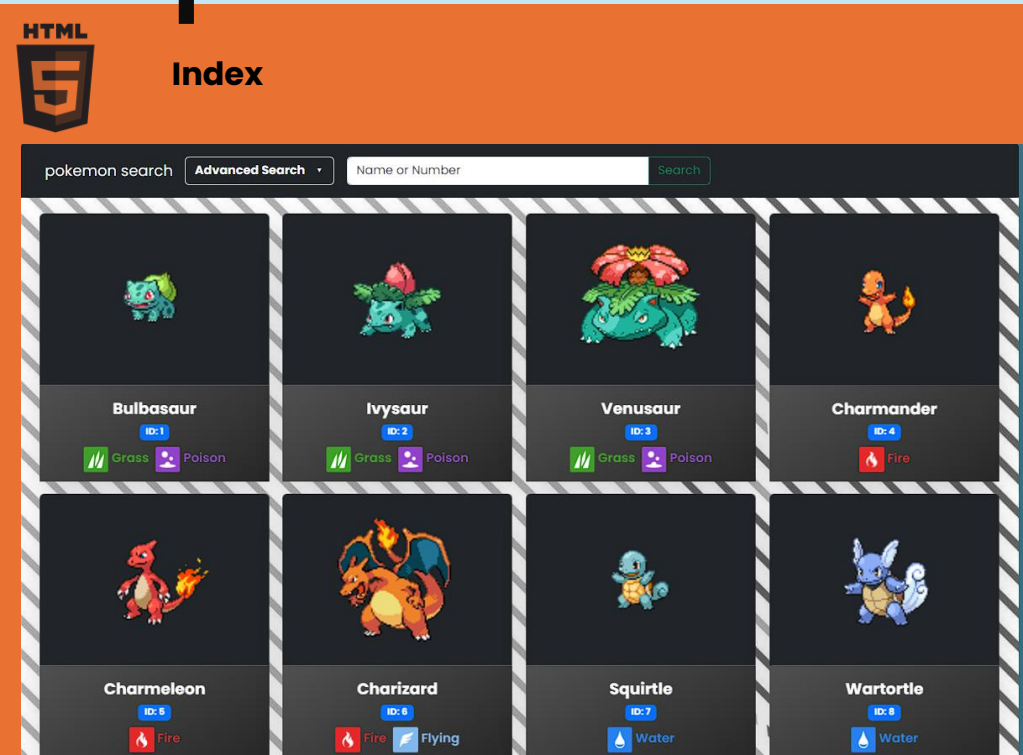
IndexedDB



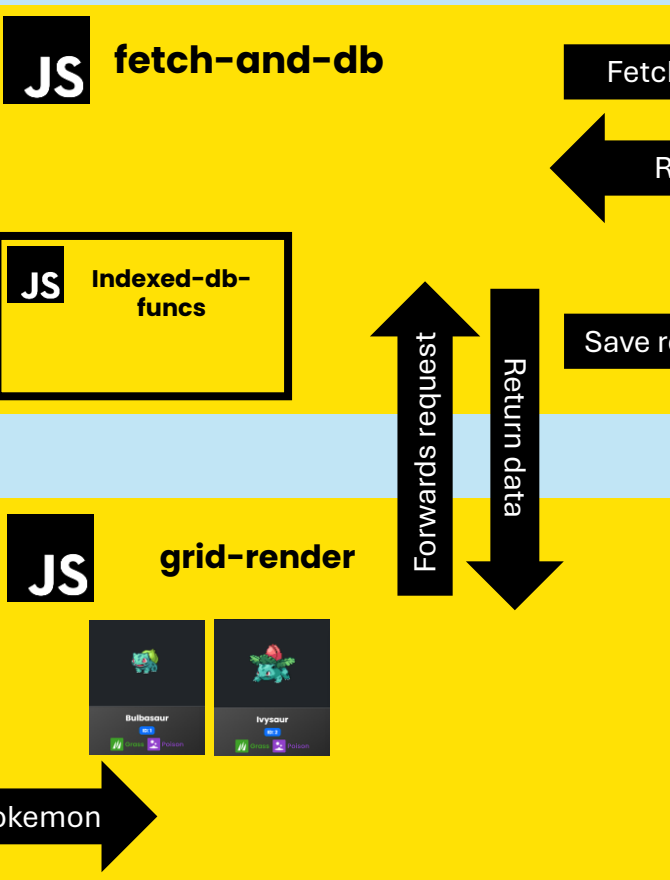


JS service-worker
JS version-manager

grid-render builds the cards in a separate fragment



Asks for 8 pokemon



Fetch from API

Response

Save response to DB

Forwards request

Return data

Dexie.js

IndexedDB



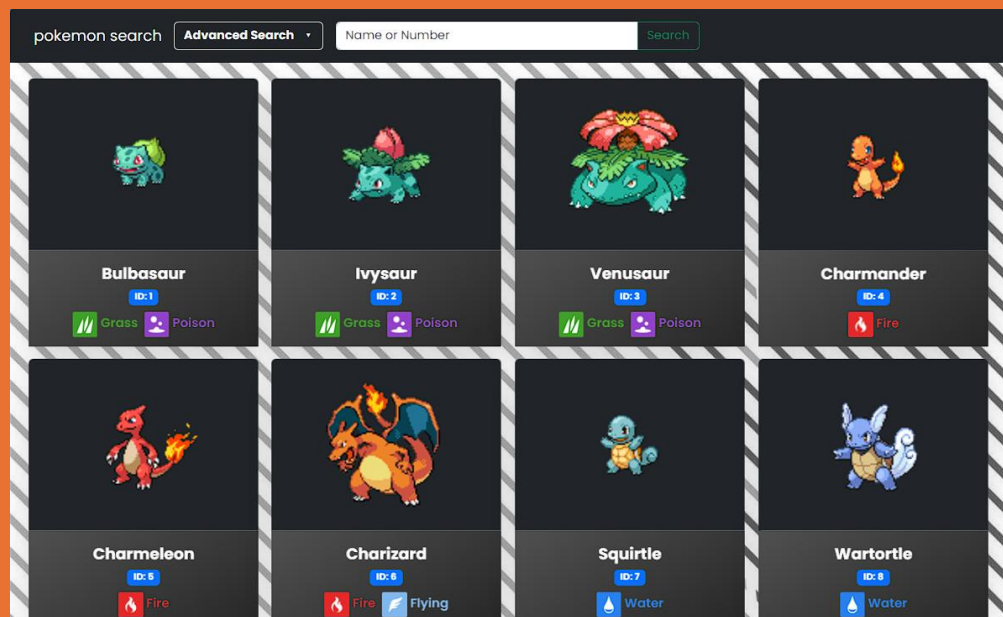


JS service-worker

JS version-manager



Index



Asks for 8 pokemon

JS fetch-and-db

Fetch from API

Response

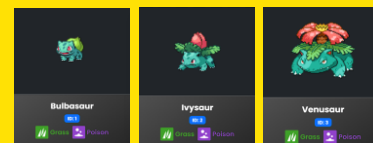
JS Indexed-db-funcs

Save response to DB

JS grid-render

Forwards request

Return data



PokeAPI

Dexie.js

IndexedDB



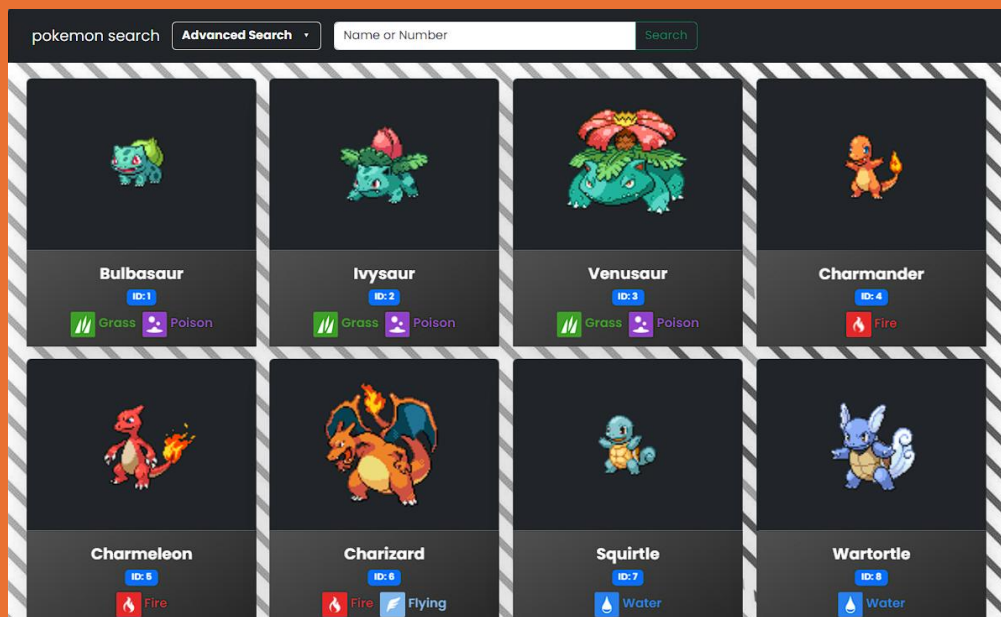


JS service-worker

JS version-manager



Index



Asks for 8 pokemon

JS fetch-and-db

Fetch from API

Response

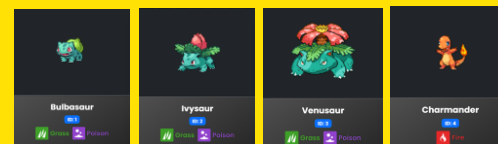
JS Indexed-db-funcs

Save response to DB

JS grid-render

Forwards request

Return data



PokeAPI

Dexie.js

IndexedDB





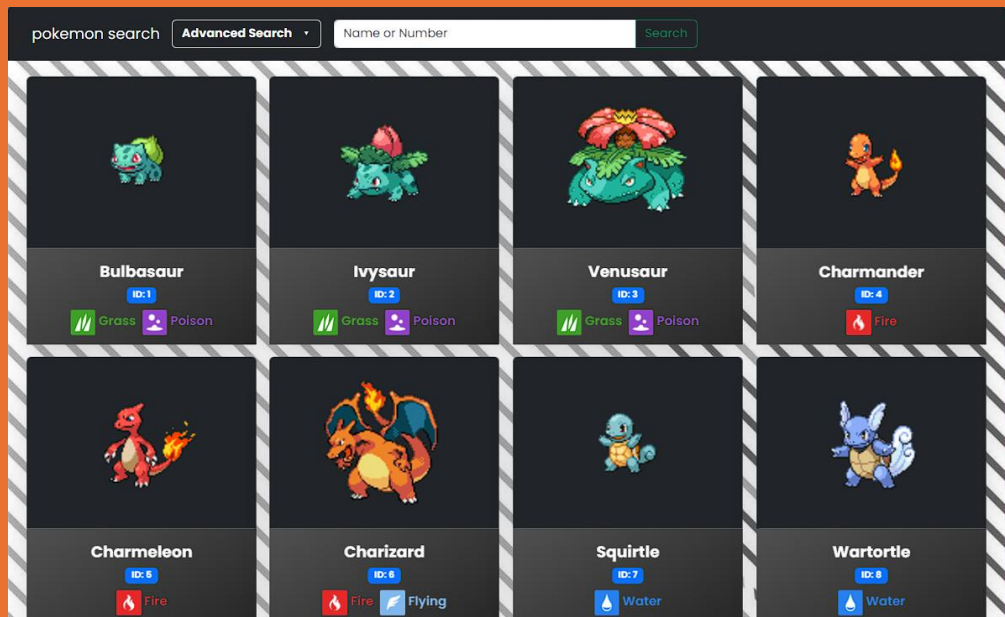
JS service-worker

JS version-manager

Items are rendered



Index



JS fetch-and-db

Fetch from API

Response

JS Indexed-db-funcs

Save response to DB

JS grid-render

Forwards request

Return data

Items are rendered

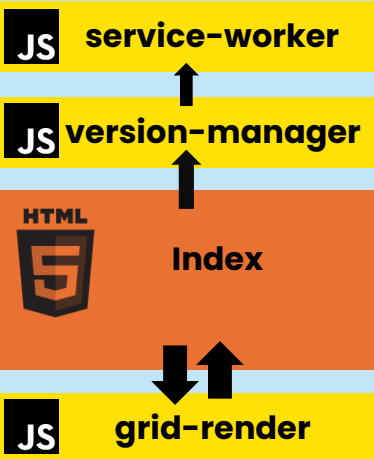
Asks for 8 pokemon

PokeAPI

Dexie.js

IndexedDB

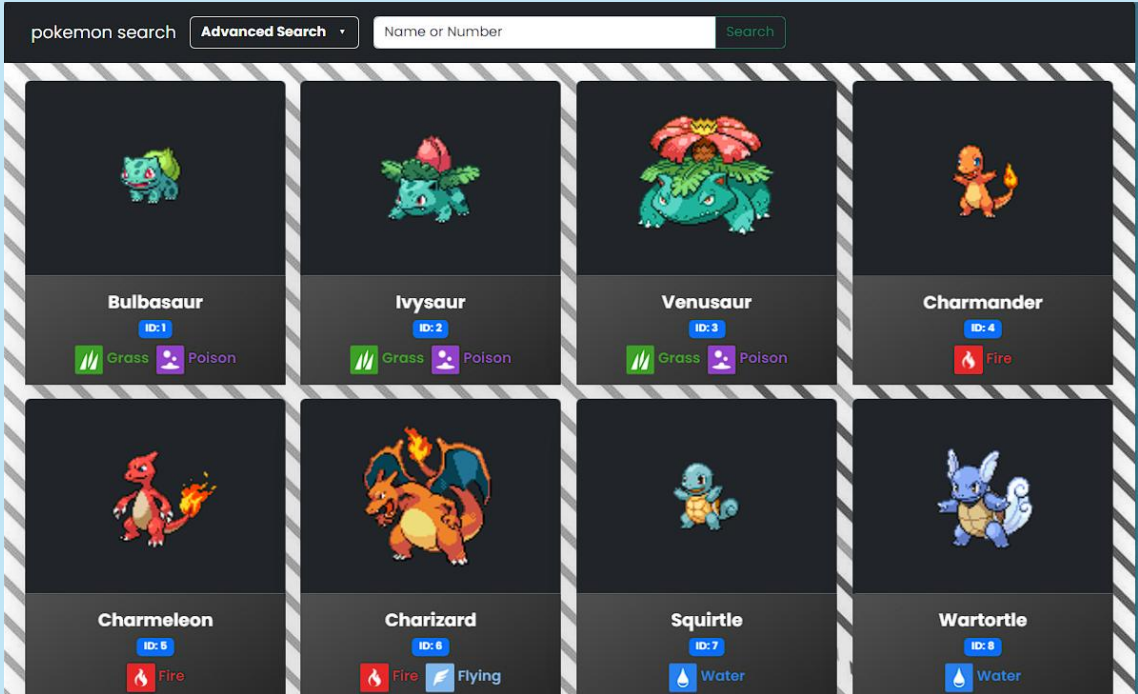




Next, we'll look at the detailed pokemon stats page

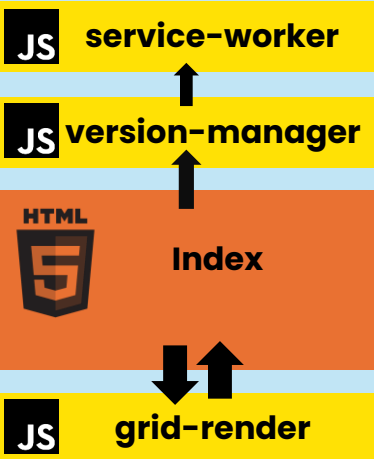
JS fetch-and-db

JS Indexed-db-funcs

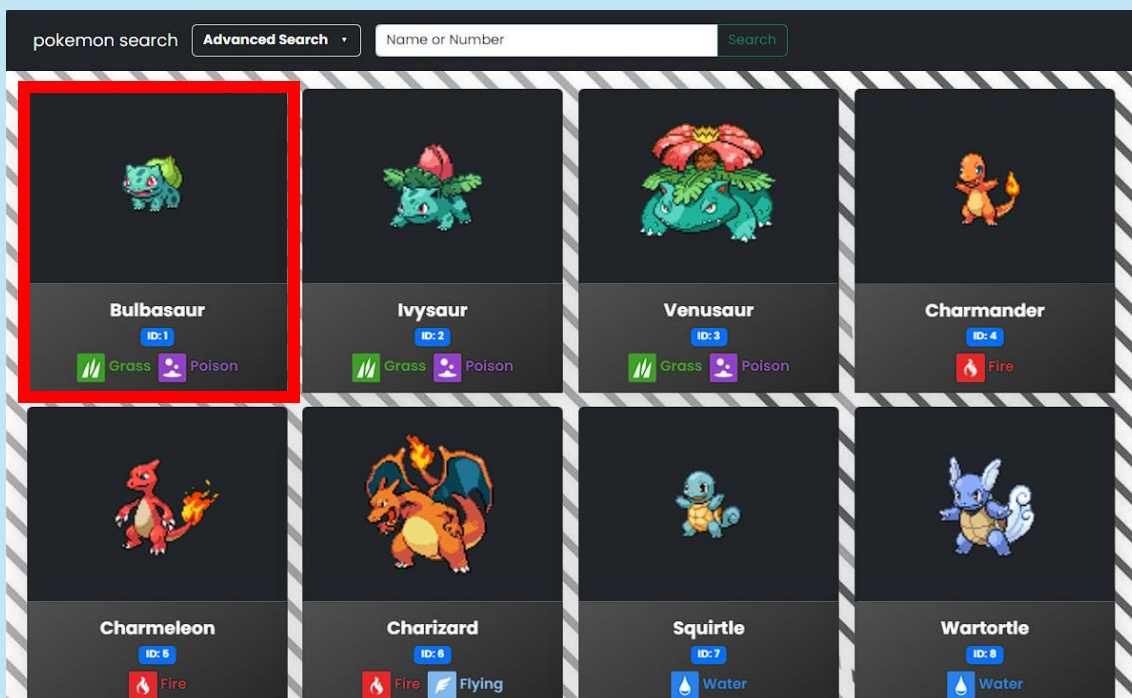
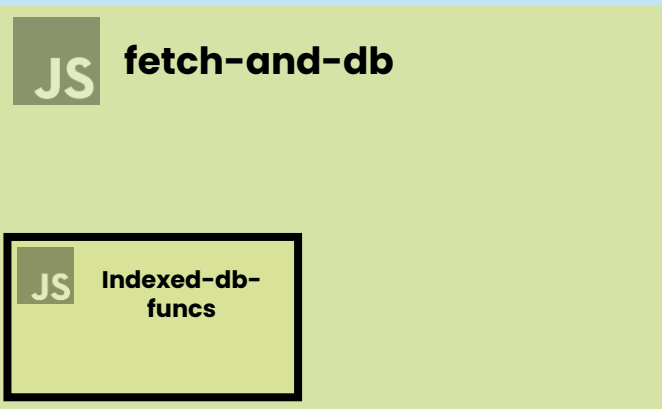


Dexie.js

IndexedDB



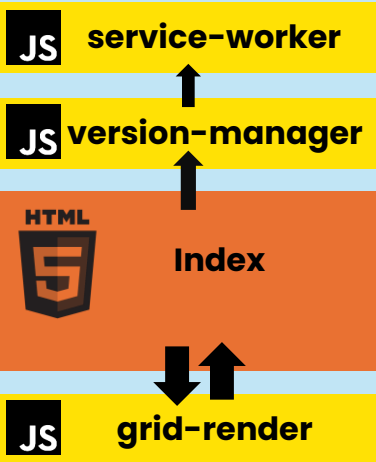
If user clicks on a card...



Dexie.js

IndexedDB

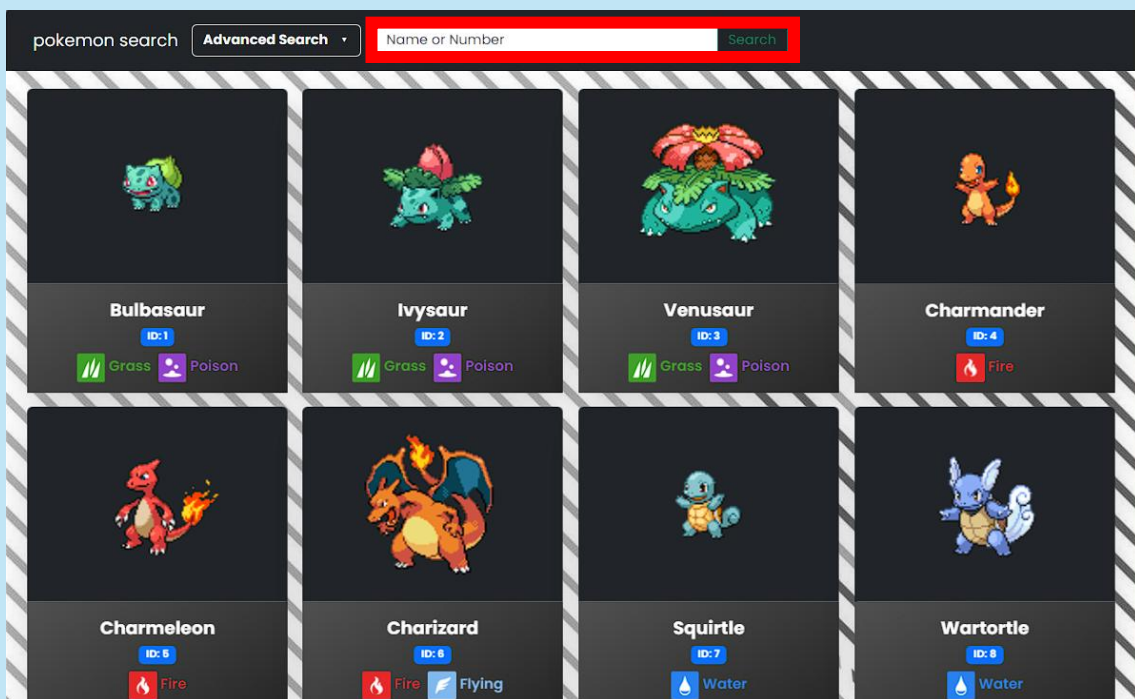




Or if user enters an ID or Name...

JS fetch-and-db

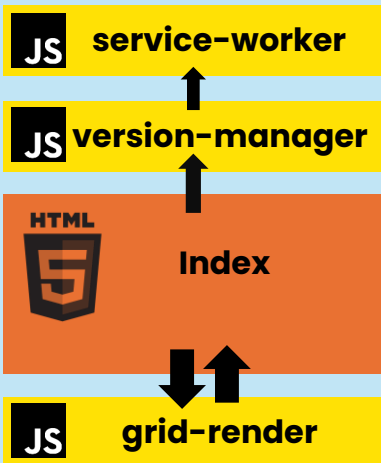
JS Indexed-db-funcs



Dexie.js

IndexedDB





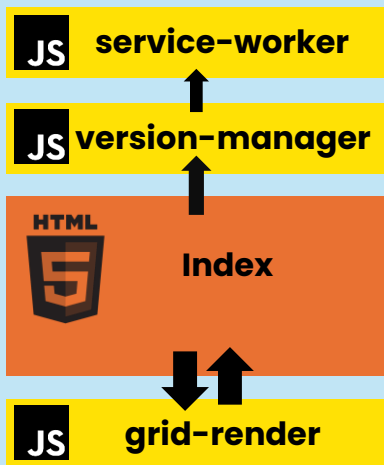
The HTML sends a GET request with the pokemon ID as parameter:

``

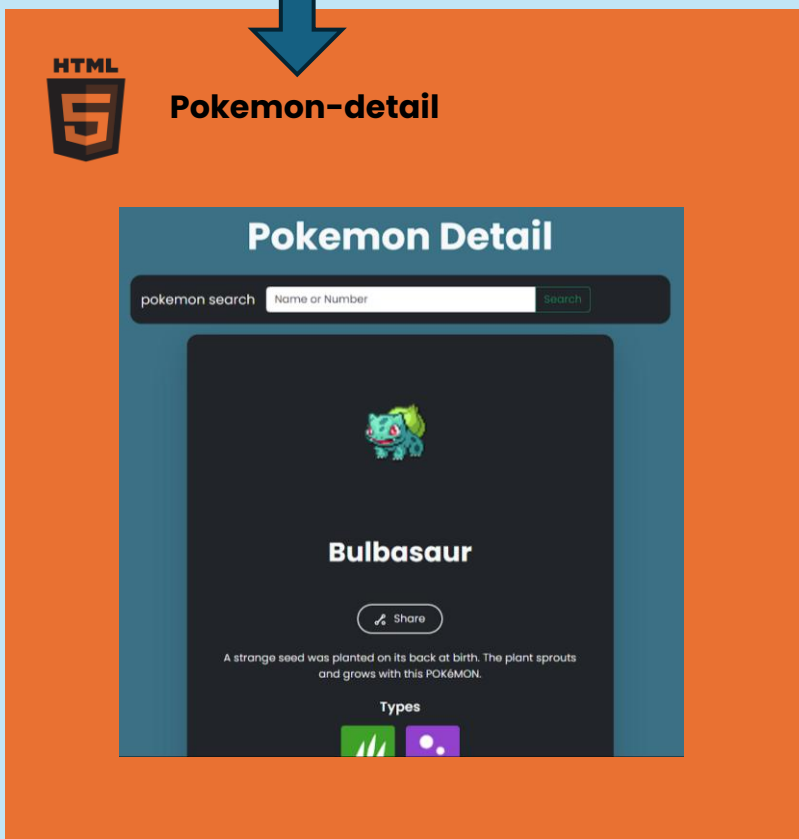




The user is sent to the page.



Generated link or
URL GET request



Dexie.js

IndexedDB





JS service-worker

JS version-manager

HTML
Index

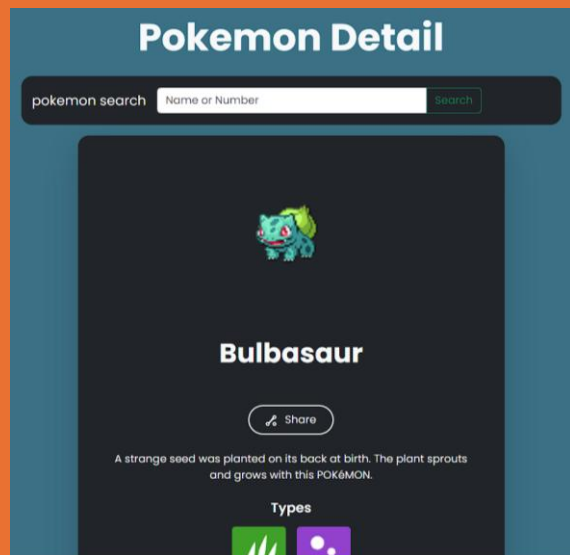
JS grid-render

We ask detail-render to create and render the elements on the page

Generated link or
URL GET request



Pokemon-detail



Asks for a pokemon

JS fetch-and-db

JS Indexed-db-funcs

JS detail-render

PokéAPI

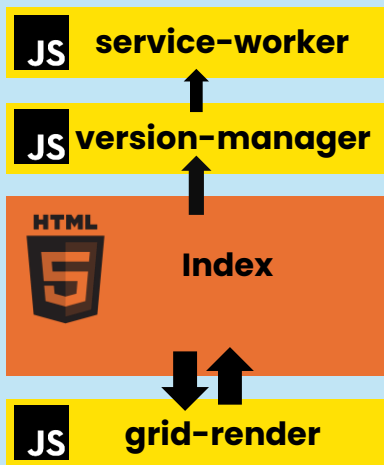
Dexie.js

IndexedDB

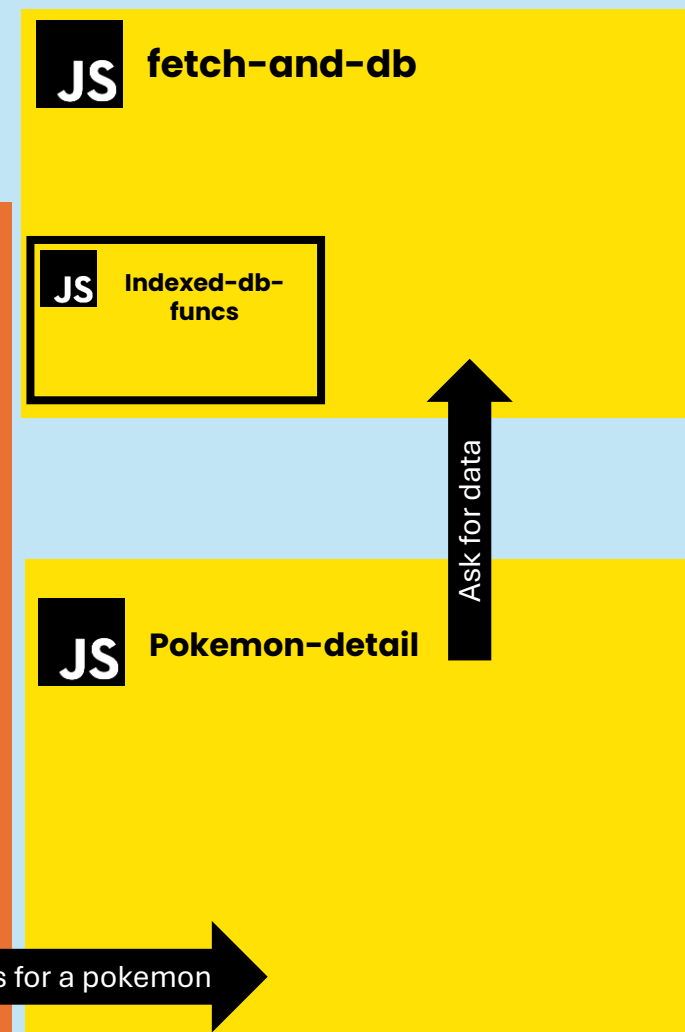
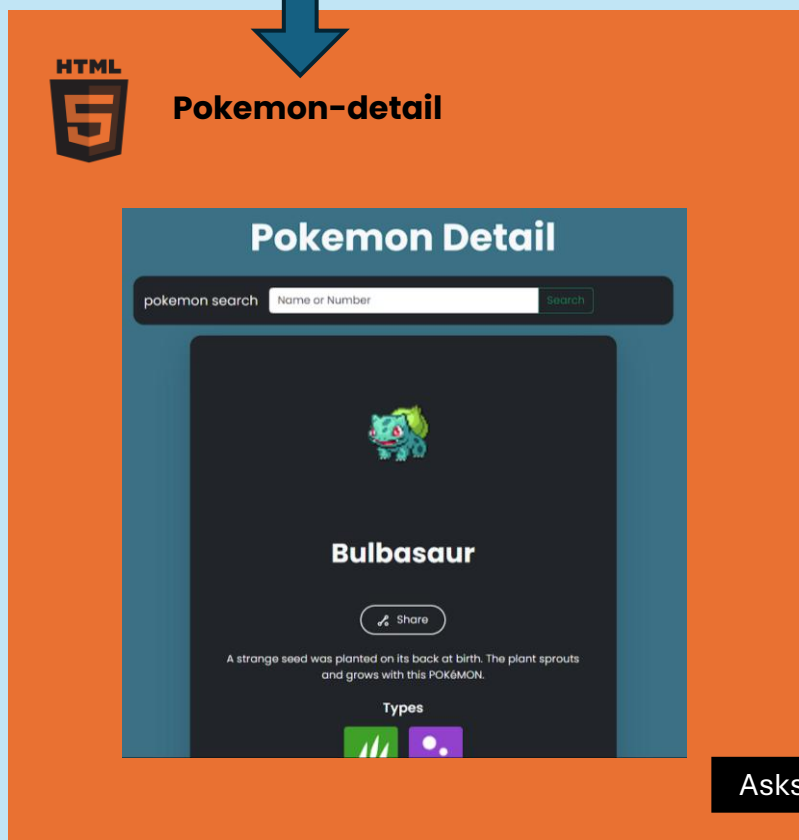




Again, we ask fetch-and-db for the json info

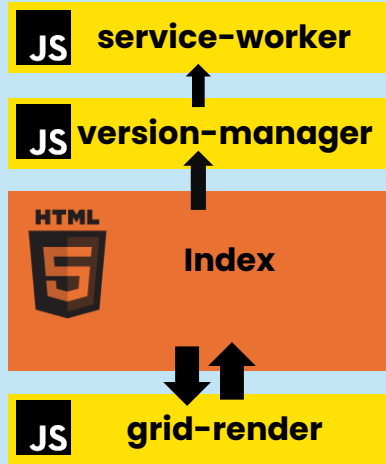


Generated link or
URL GET request






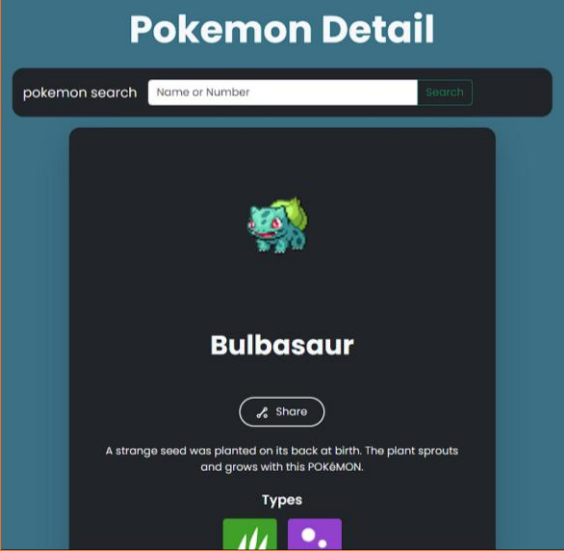
We find it in the local DB this time.



Generated link or URL GET request



**Pokemon-detail**



JS fetch-and-db

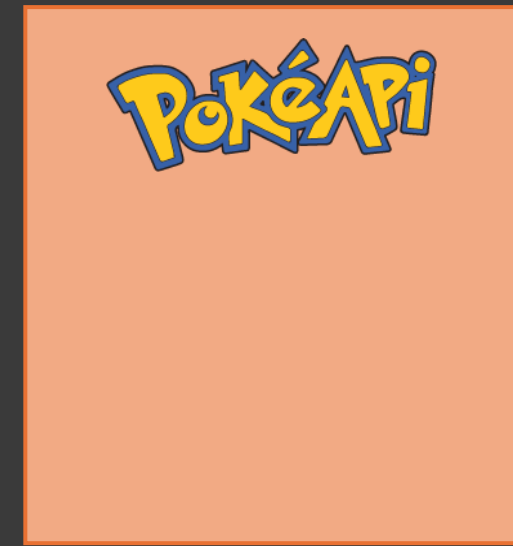
JS Indexed-db-funcs

Check DB →

JS detail-render

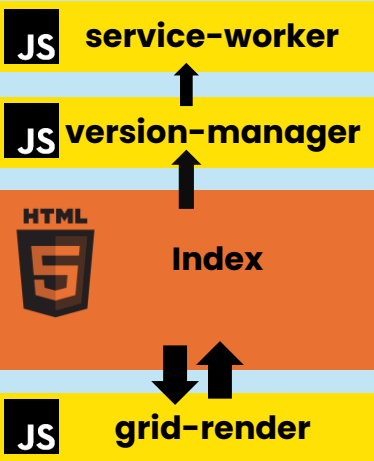
↑ Ask for data

→ Asks for a pokemon



**Dexie.js**

IndexedDB

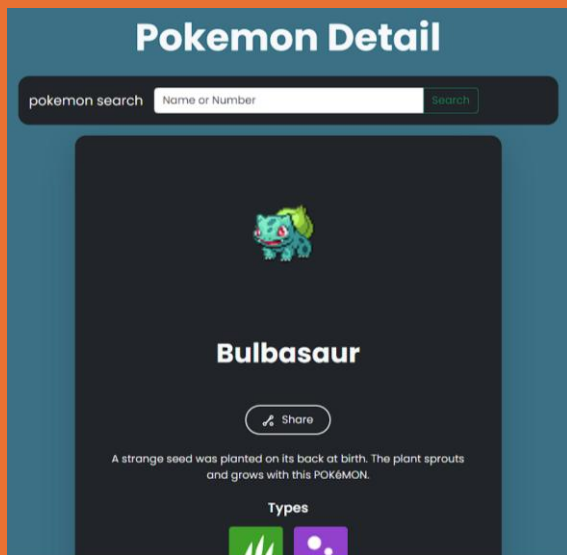



Return data

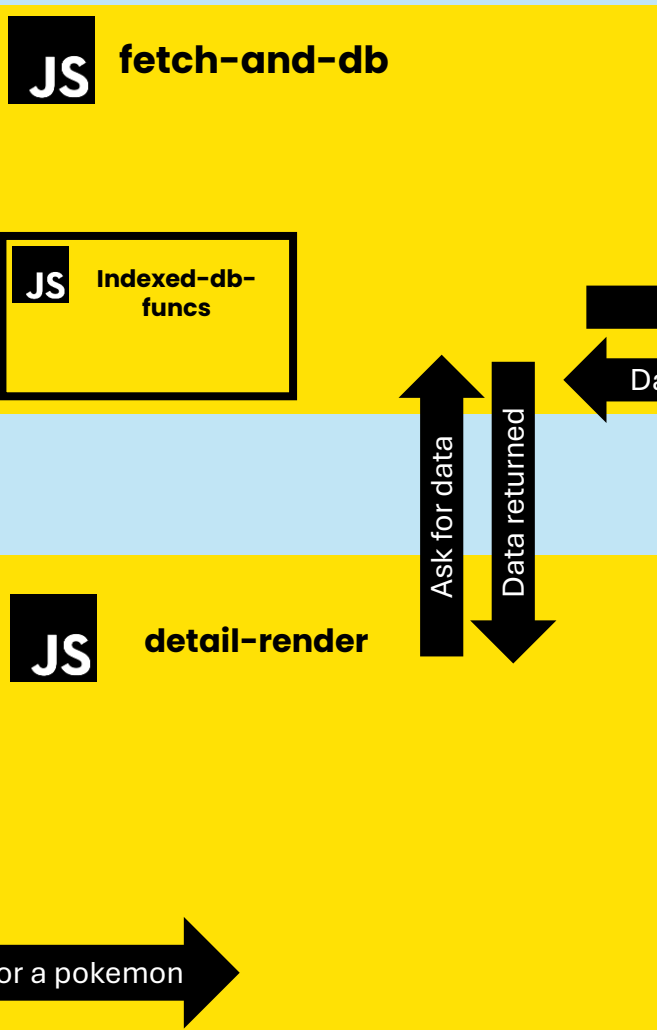
Generated link or URL GET request



Pokemon-detail



Asks for a pokemon



Check DB

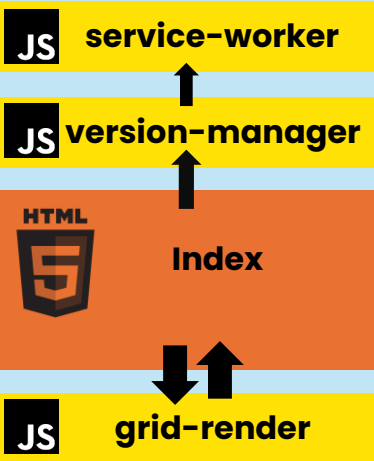
Data returned

✓ Data found

Dexie.js

IndexedDB



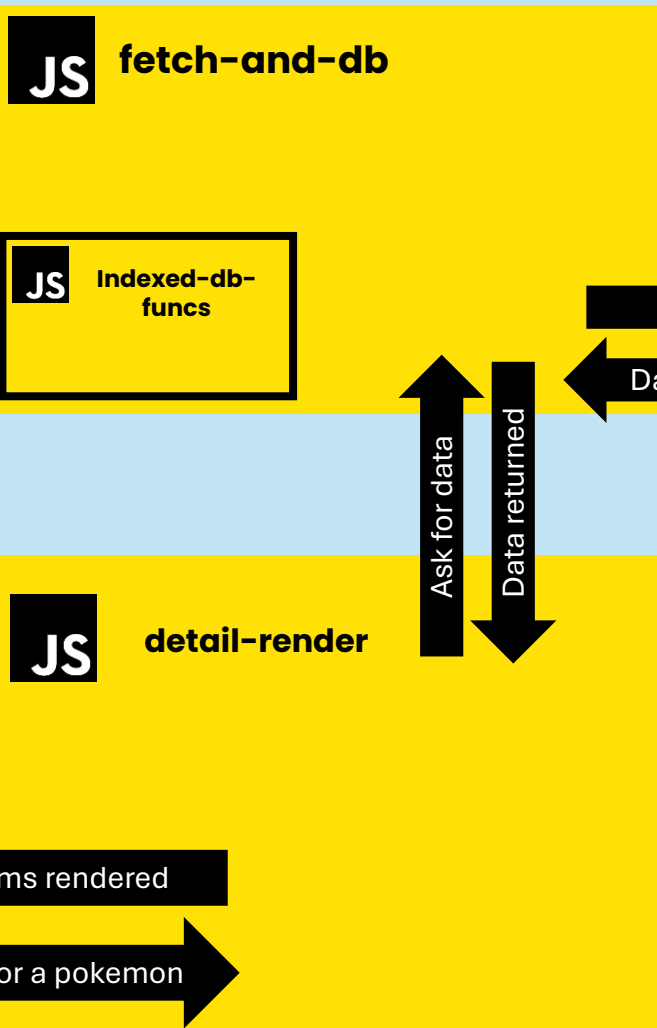
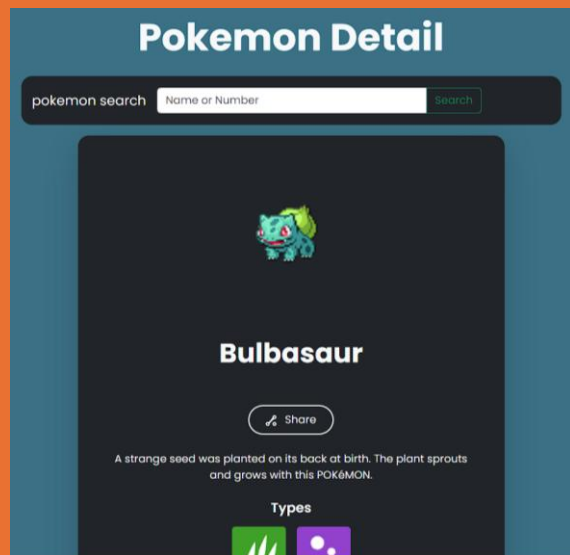


Render items

Generated link or
URL GET request



Pokemon-detail



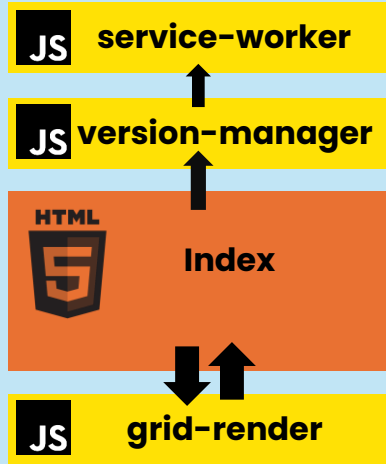
Dexie.js

IndexedDB





The D3 chart is created by detail-render-chart



Generated link or URL GET request

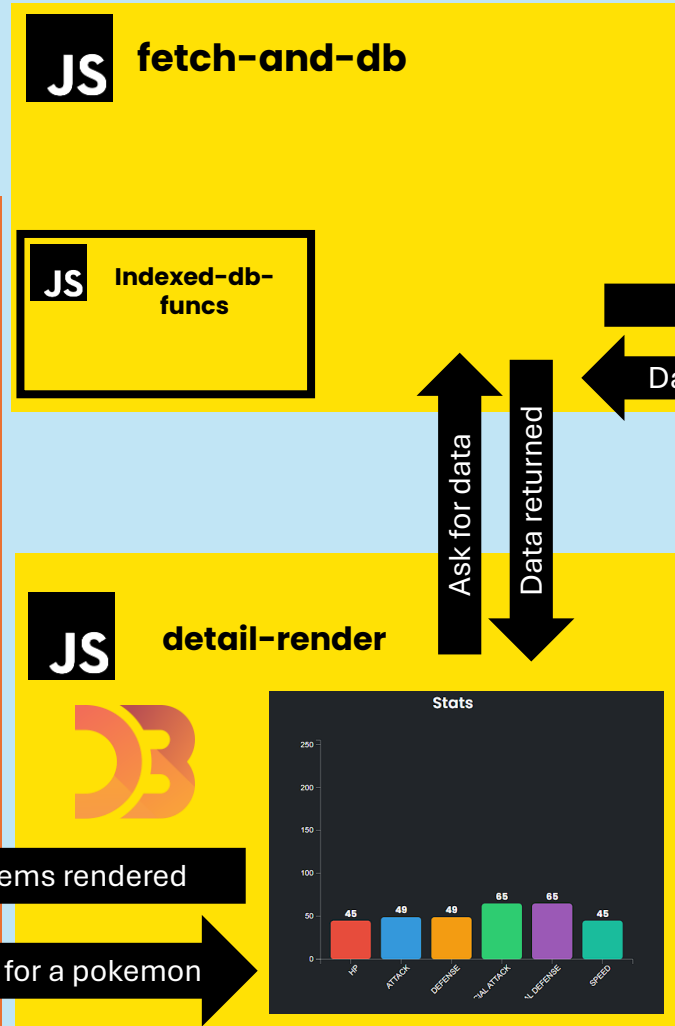
Pokemon Detail

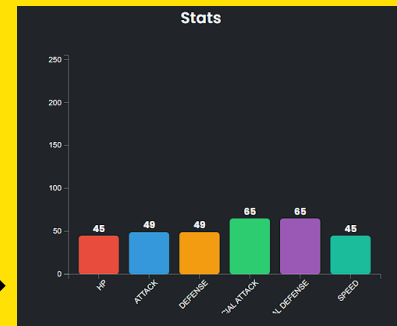
pokemon search

Bulbasaur

A strange seed was planted on its back at birth. The plant sprouts and grows with this POKEMON.

Types







Simplified vue

