



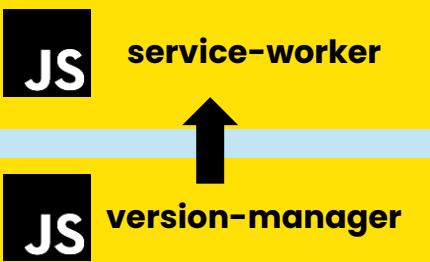




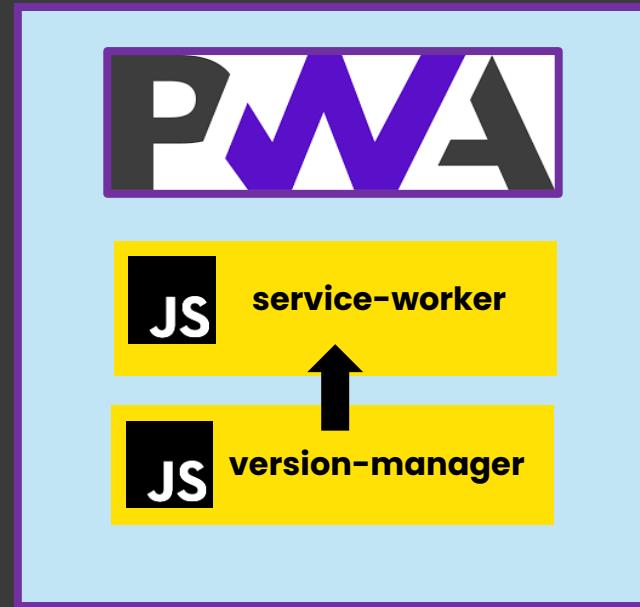
JS service-worker

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

JS version-manager



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





JS service-worker



JS version-manager



JS service-worker



JS version-manager



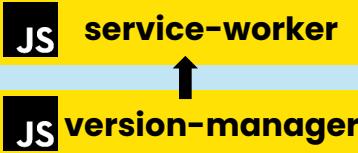
User is presented with Index

Index

Welcome to PWAkedex

pokemon search [Advanced 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



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

HTML 5

The screenshot shows the PWAkedex app's index page. At the top, there is a navigation bar with links for Home, Pokemon, Games, and Items. On the right side of the header, it says "Pokémon in cache: 1350/1350" and has a "Install PWA?" button. Below the header, there is a search bar with fields for "pokemon search" and "Advanced Search", and a "Name or Number" input field with a "Search" button. The main content area displays a grid of eight initial generation Pokémon cards:

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

A modal window titled "Install as PWA?" is centered over the grid. It contains the text "Installing the app as a PWA allows for offline use" and two buttons: "Install" and "Close".

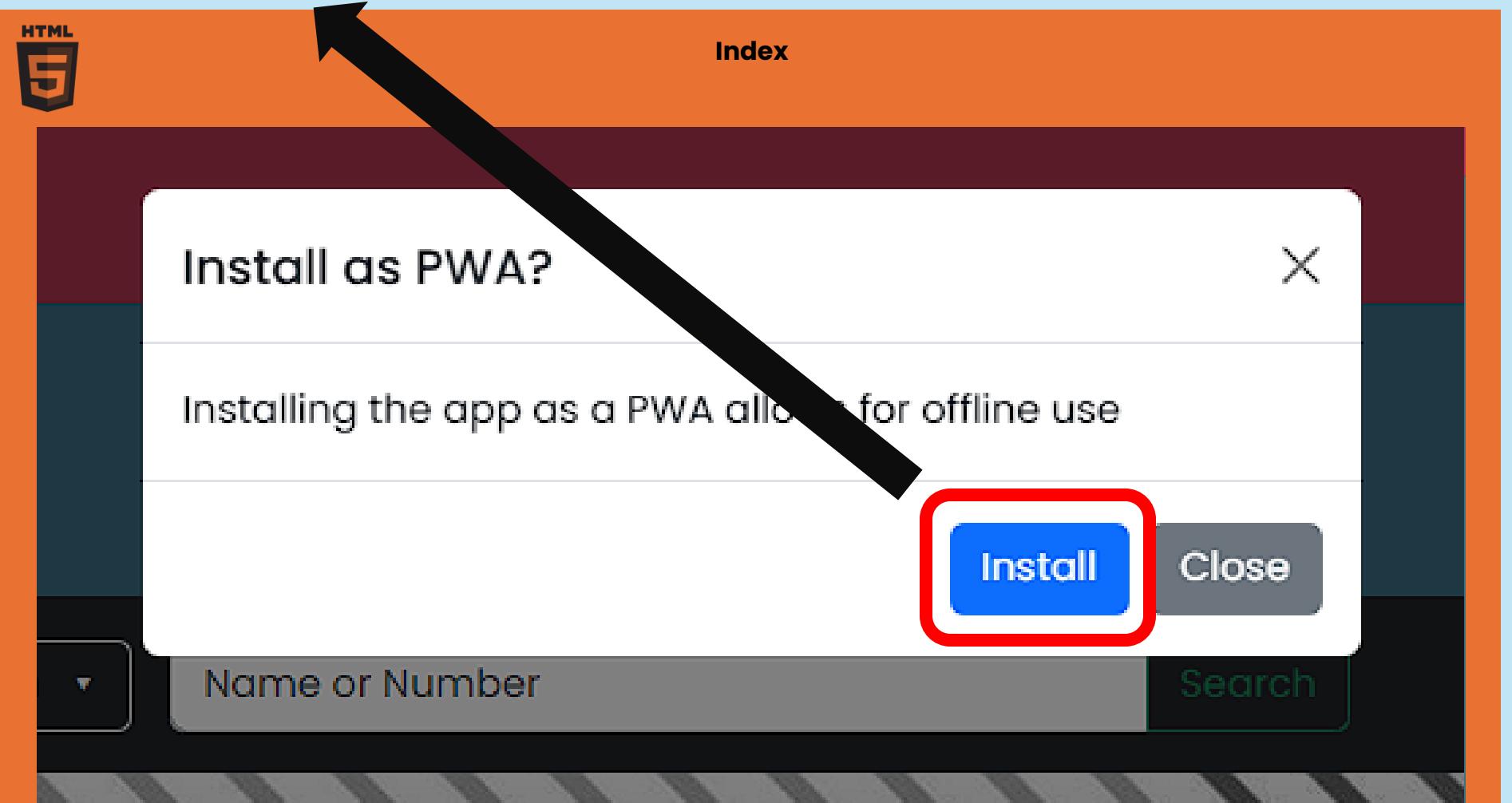
PWA

JS service-worker

JS version-manager

HTML
5

If accepted, version-manager installs the PWA





JS service-worker

JS version-manager

HTML
5

If accepted, version-manager installs the PWA

Index

PWAkedex Home Pokémon Games Items

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

Welcome to PWAkedex

pokemon search [Advanced Search](#) Name or Number [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

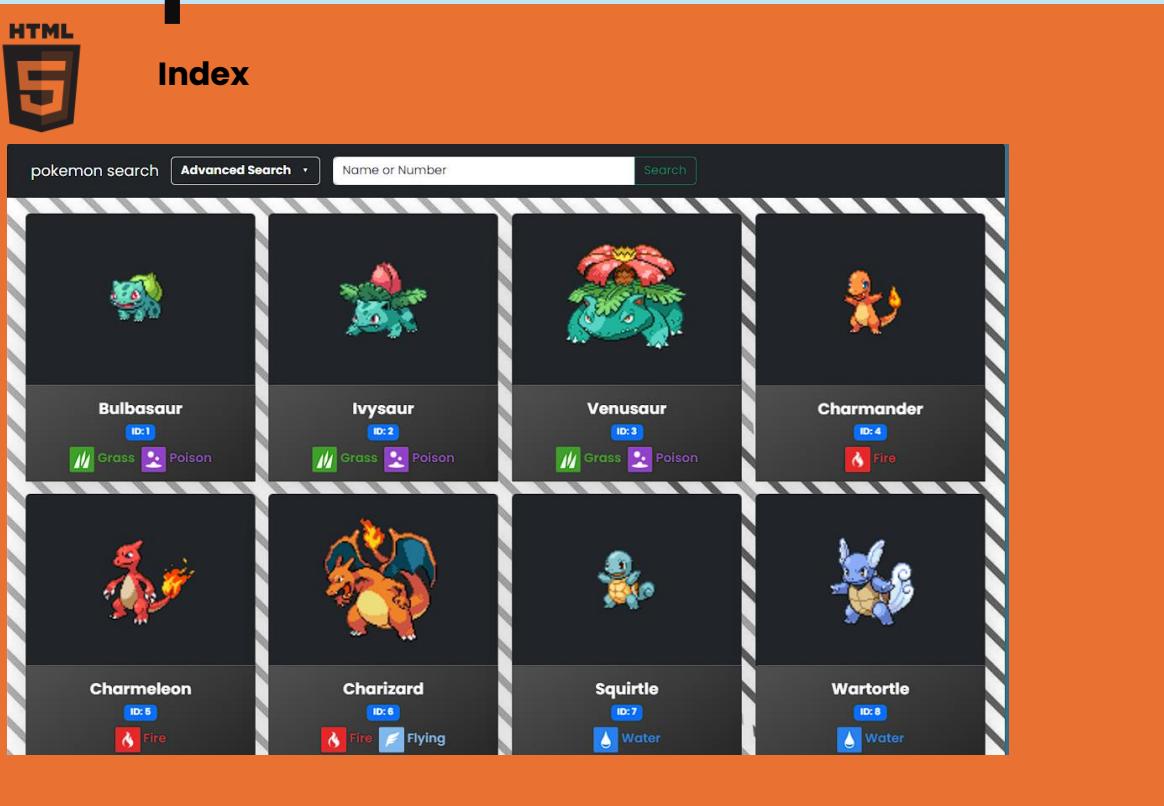


JS service-worker

JS version-manager

HTML
5
Index

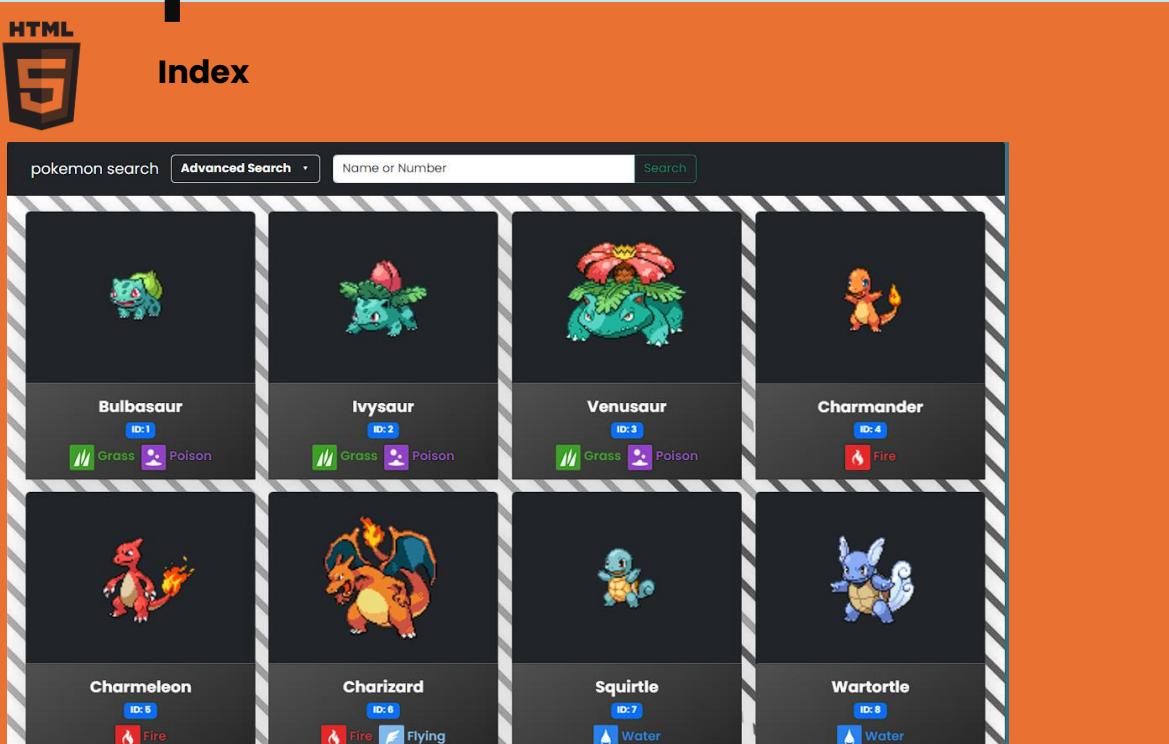
The grid of pokemon is rendered by grid-render

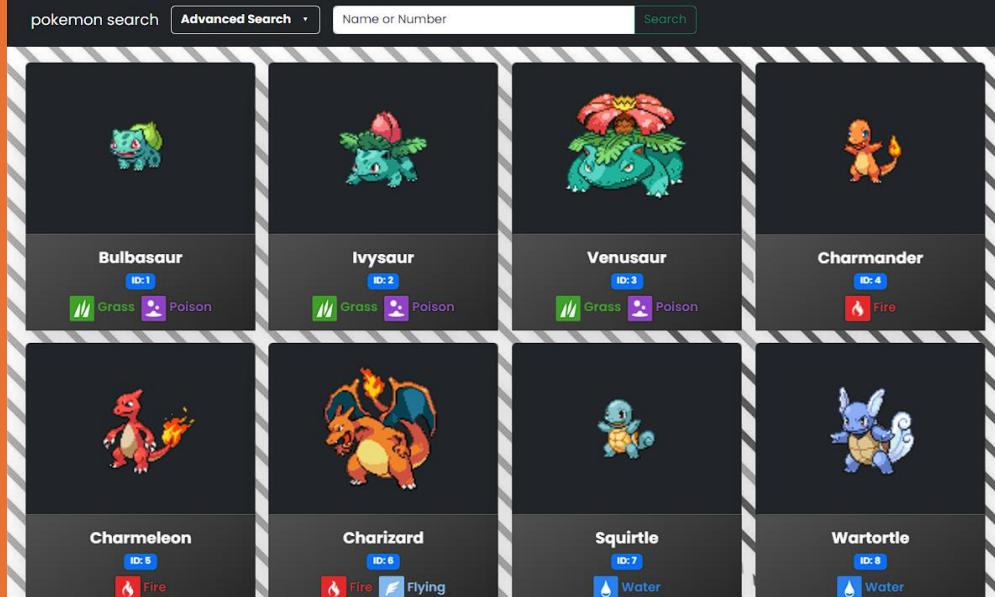
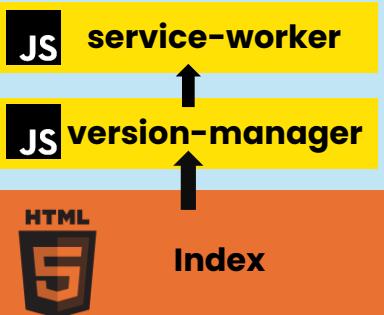


JS grid-render

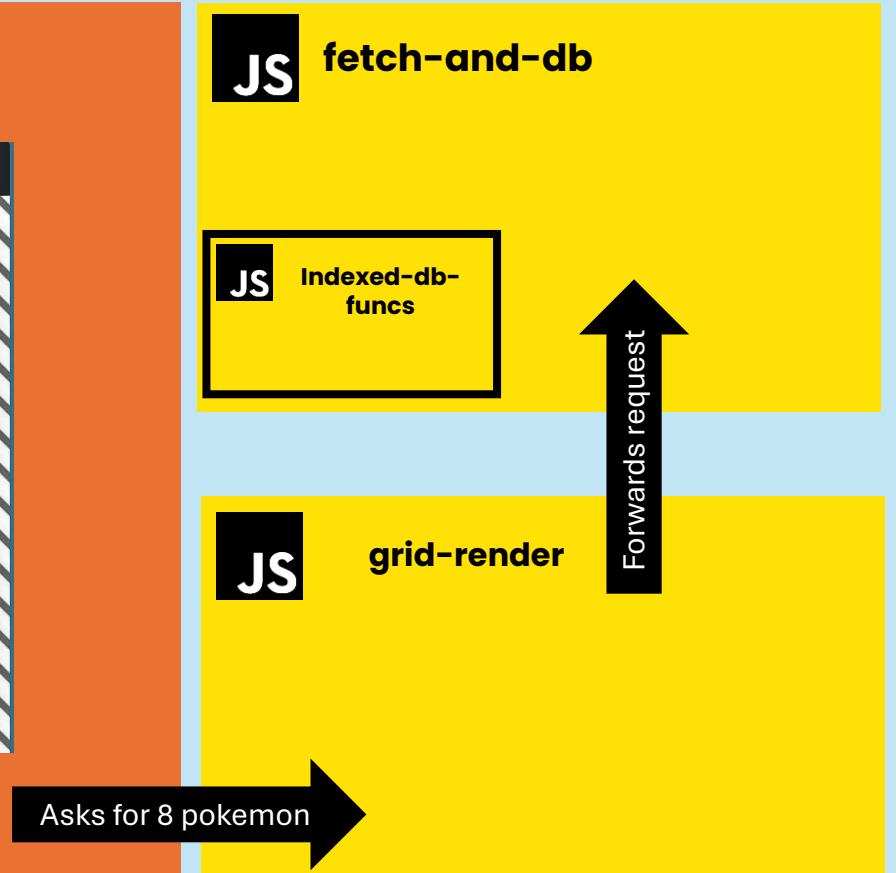


On page load, grid-render is asked to render 8 cards*





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

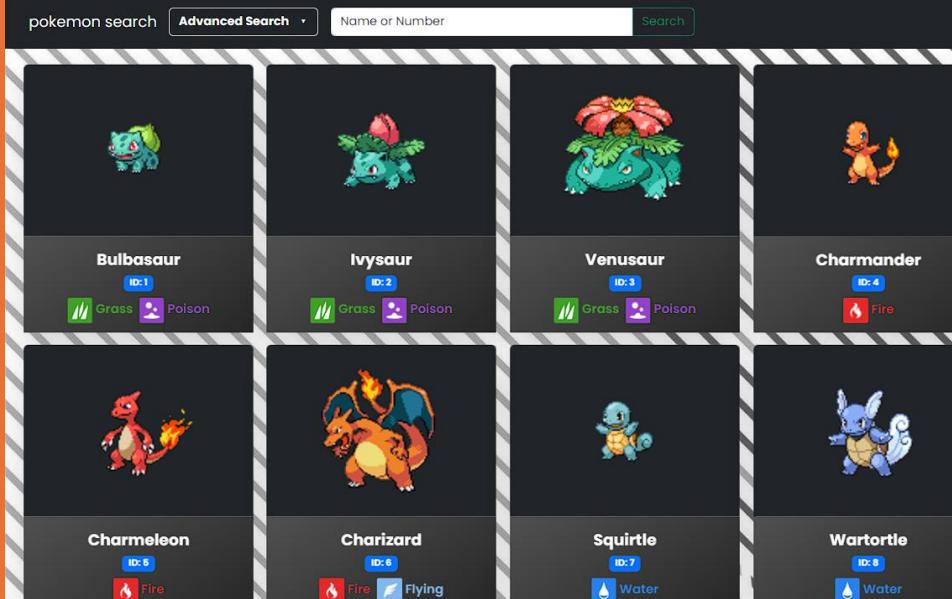


PWA

JS service-worker

JS version-manager

HTML
5
Index



Data can come from the API or cache

JS fetch-and-db

JS Indexed-db-funcs

JS grid-render

Asks for 8 pokémon

Forwards request

PokeAPI

Dexie.js

IndexedDB



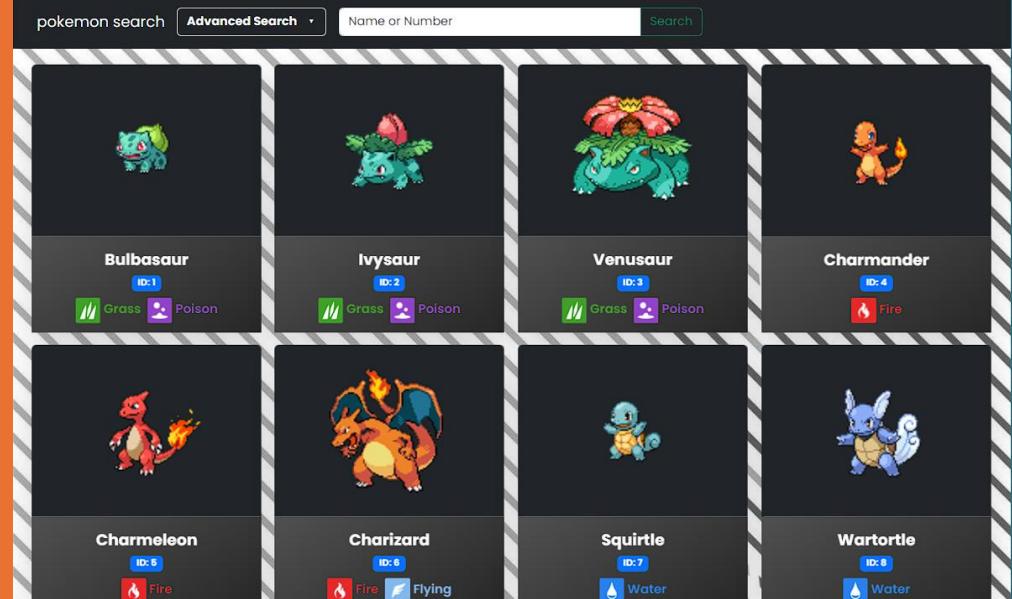
Dexie.js internal API handles Indexed DB requests for cleaner code

PWA

JS service-worker

JS version-manager

HTML
5
Index



First, we check if we have the json in cache

JS fetch-and-db

JS Indexed-db-funcs

JS grid-render

Asks for 8 pokémon

Checks cache

Forwards request

PokeAPI

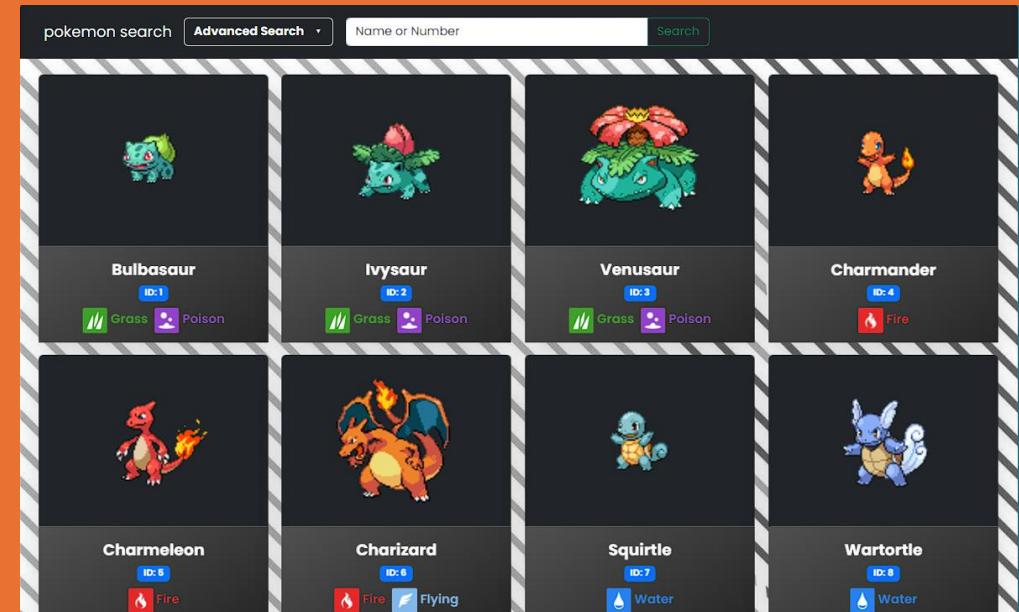
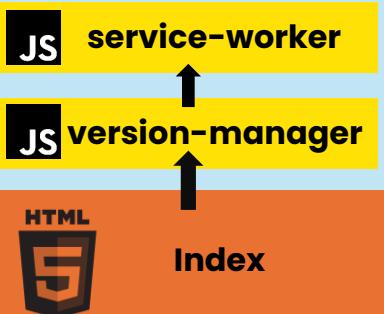
Dexie.js

IndexedDB



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

PWA



Asks for 8 pokémon

JS fetch-and-db

JS Indexed-db-funcs

JS grid-render

Checks cache

Forwards request

NOT FOUND

Dexie.js

IndexedDB



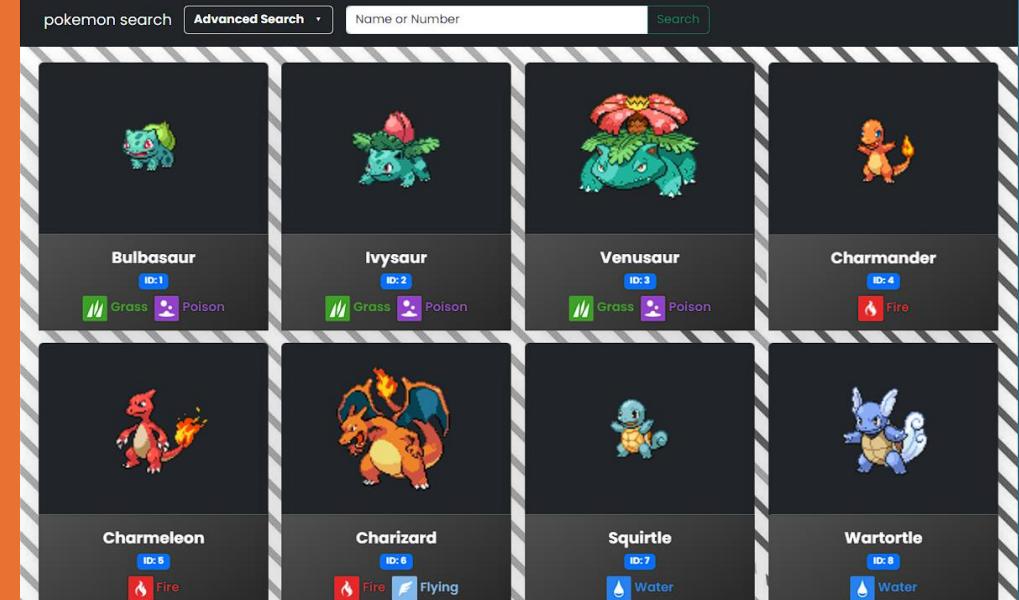
Dexie.js internal API handles Indexed DB requests for cleaner code

PWA

JS service-worker

JS version-manager

HTML
5
Index



If not, we ask the API

JS fetch-and-db

JS Indexed-db-funcs

JS grid-render

Asks for 8 pokémon

Fetch from API

Forwards request

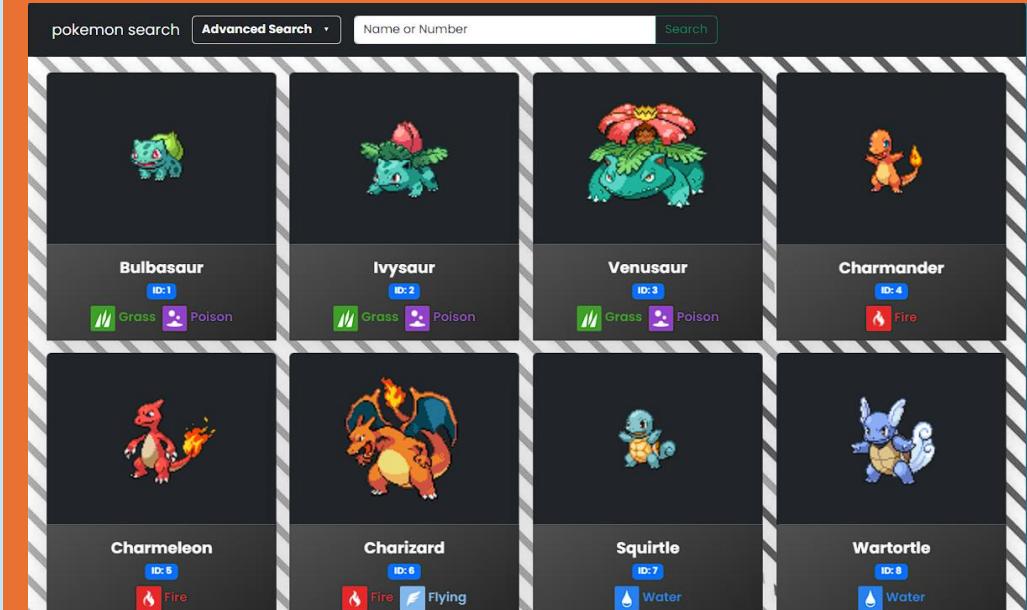
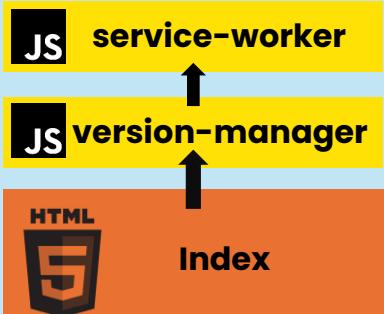
PokeAPI

Dexie.js

IndexedDB



PWA



Data is passed back to fetch-and-db

JS Fetch-and-DB

JS Indexed-db-funcs

JS grid-render

Asks for 8 pokemon

Fetch from API
Response

Forwards request

Dexie.js

IndexedDB



PokéAPI

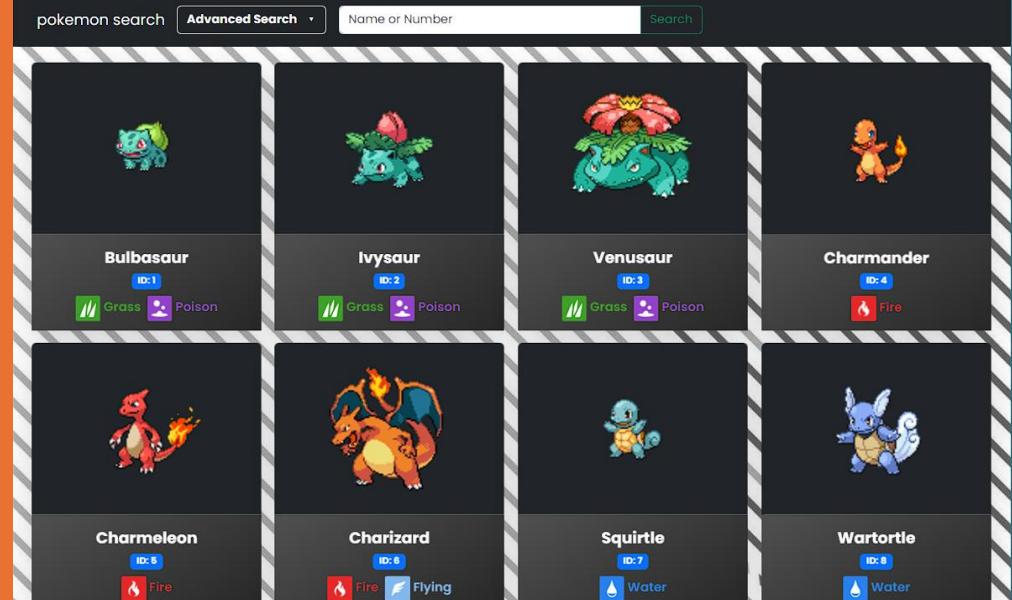


PWA

JS service-worker

JS version-manager

HTML
5
Index



fetch-and-db saves the json to indexedDB

JS fetch-and-db

JS Indexed-db-funcs

JS grid-render

Asks for 8 pokémon

Fetch from API
Response

Save response to DB

Dexie.js

IndexedDB



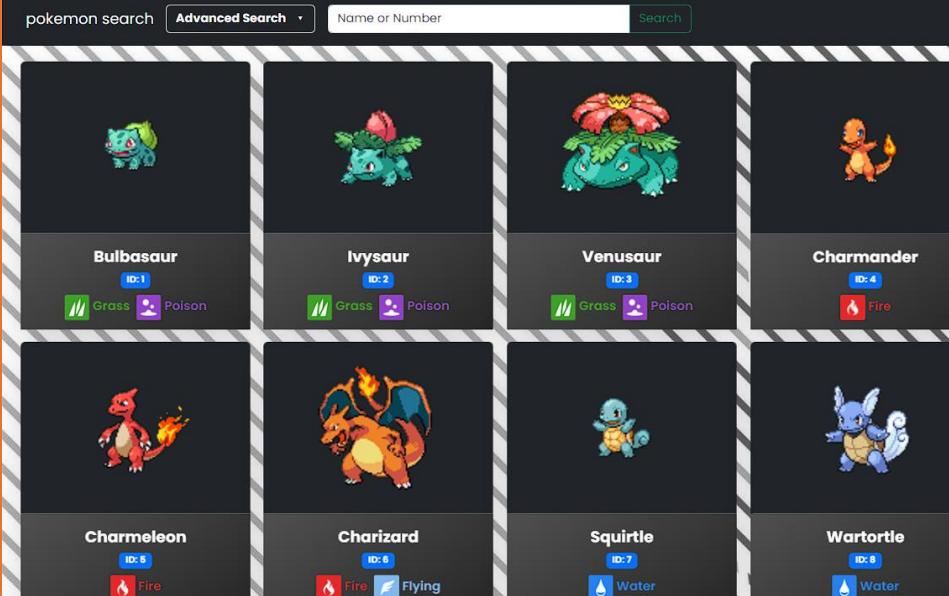
Forwards request

PWA

JS service-worker

JS version-manager

HTML
5
Index



Data is passed back to grid-render

JS fetch-and-db

JS Indexed-db-funcs

JS grid-render

Asks for 8 pokémon

Fetch from API
Response

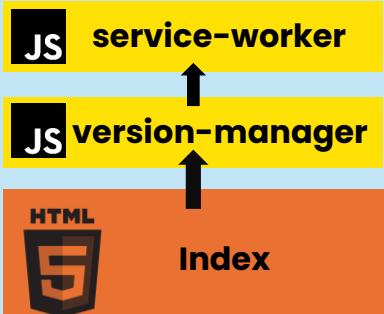
Save response to DB

Forwards request
Return data

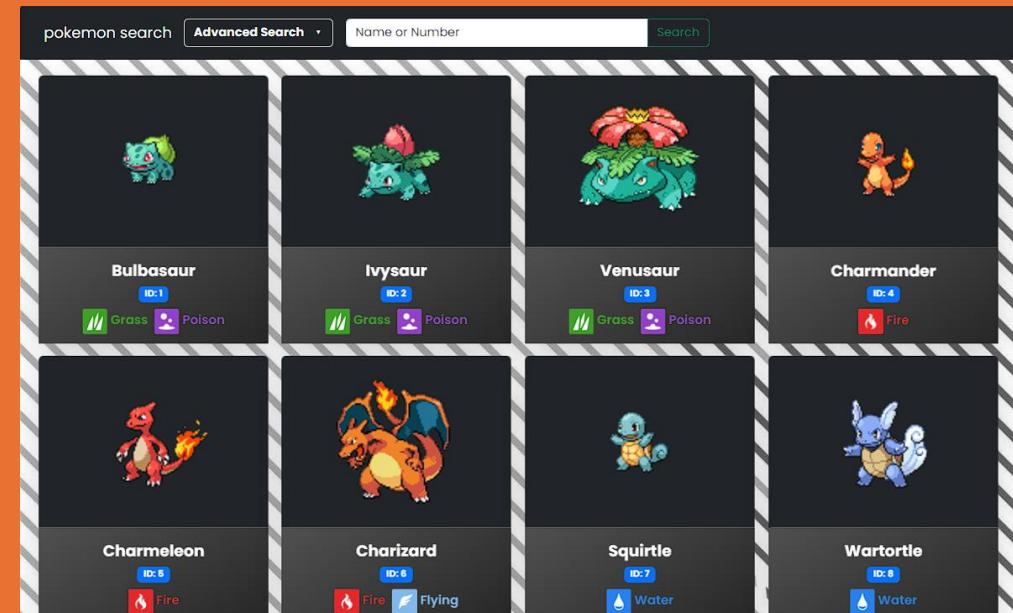
Dexie.js

IndexedDB

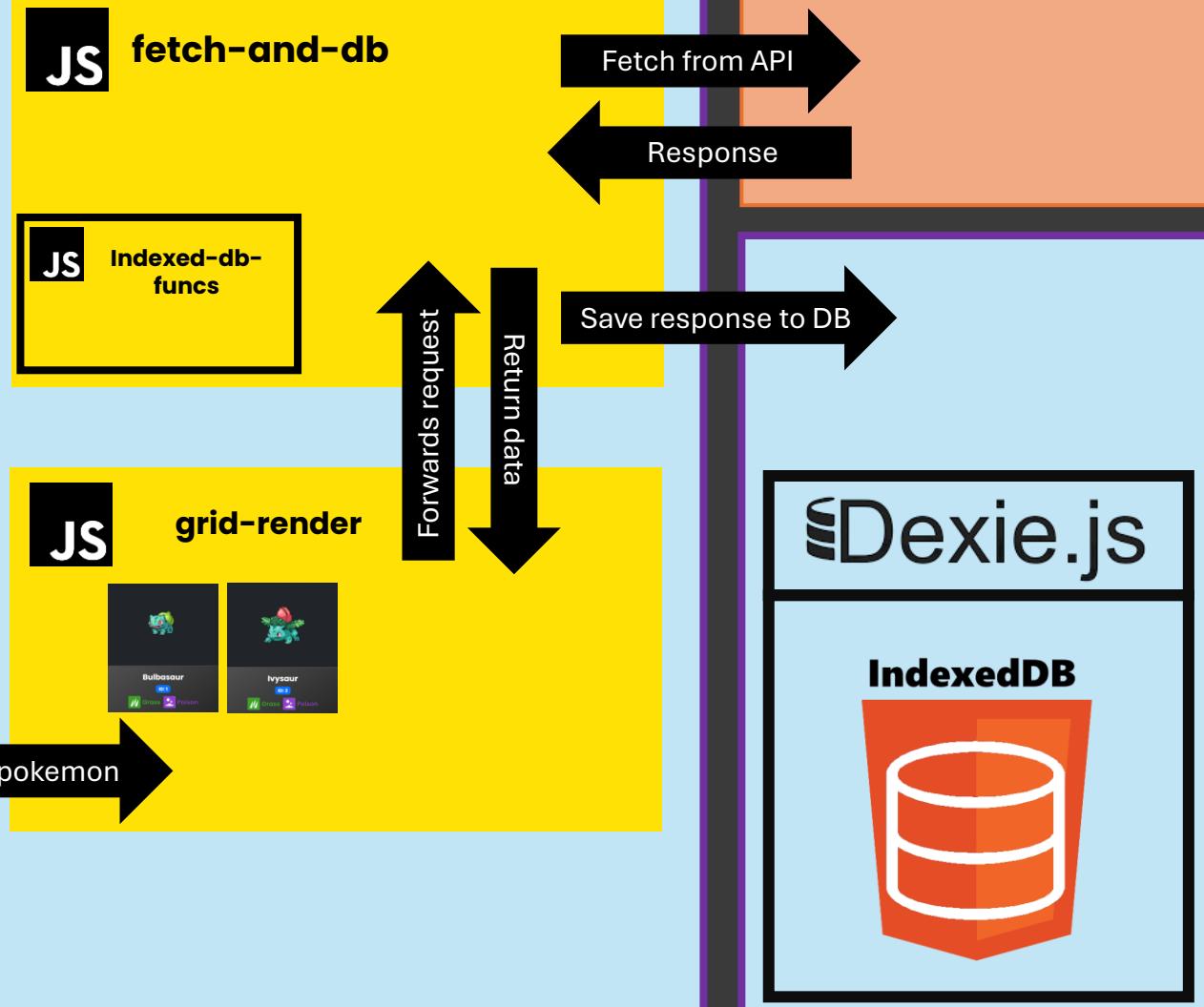




grid-render builds the cards in a separate fragment



Asks for 8 pokemon

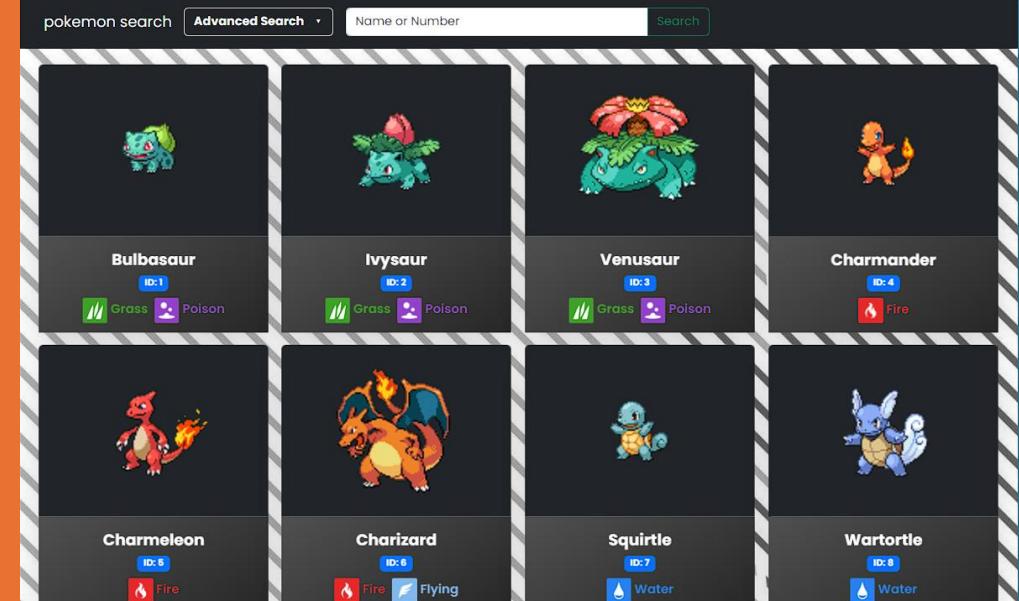


PWA

JS service-worker

JS version-manager

HTML
5
Index



Asks for 8 pokémon

JS fetch-and-db

JS Indexed-db-funcs

JS grid-render

Fetch from API
Response

Save response to DB

Forwards request
Return data

Dexie.js

IndexedDB

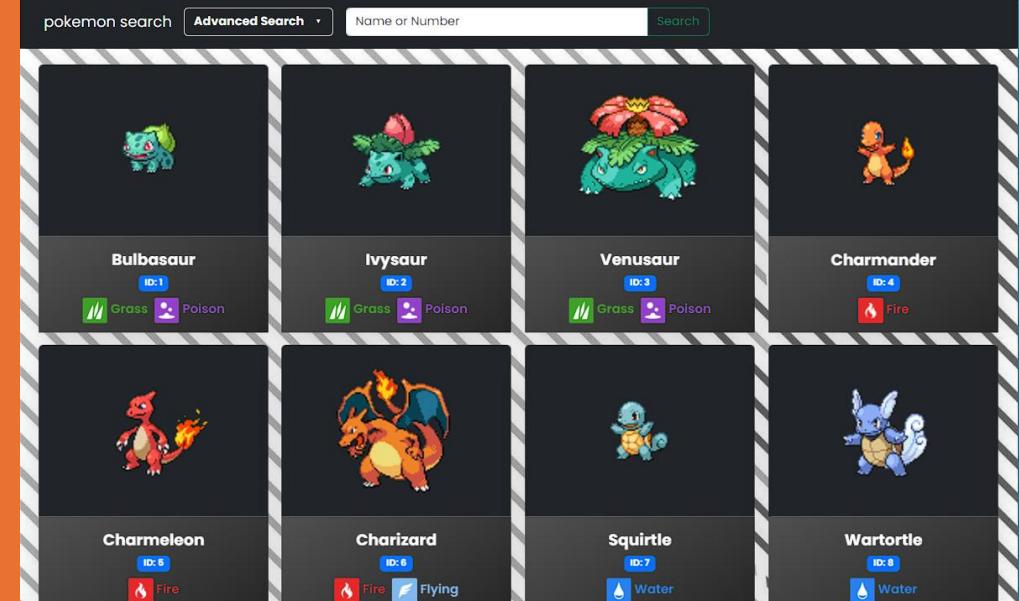


PWA

JS service-worker

JS version-manager

HTML
5
Index



Asks for 8 pokémon

JS fetch-and-db

JS Indexed-db-funcs

JS grid-render

Fetch from API

Response

Save response to DB

Forwards request
Return data

Dexie.js

IndexedDB

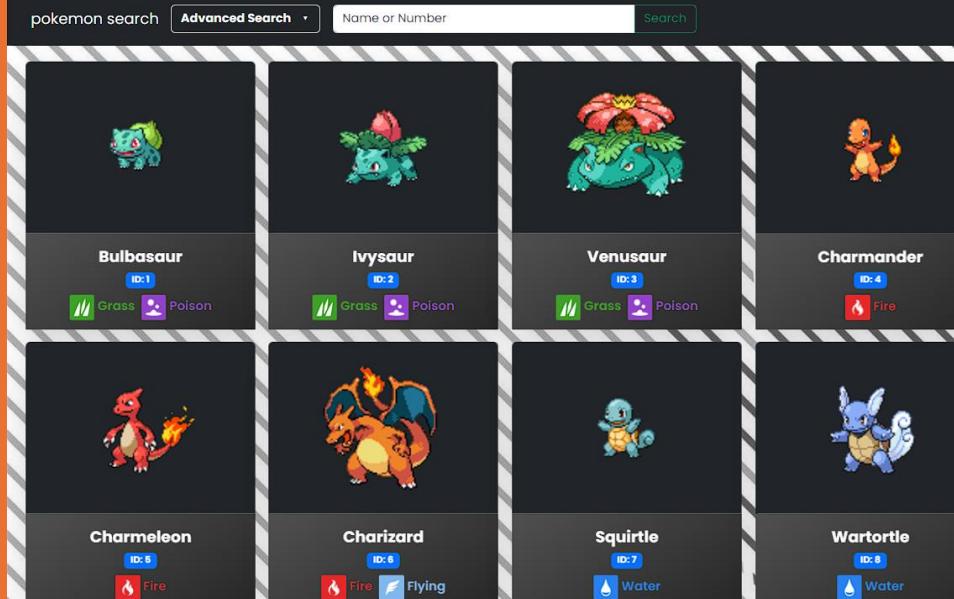




JS service-worker

JS version-manager

HTML
5
Index



Items are rendered

JS fetch-and-db

JS Indexed-db-funcs

JS grid-render

Fetch from API
Response
Save response to DB

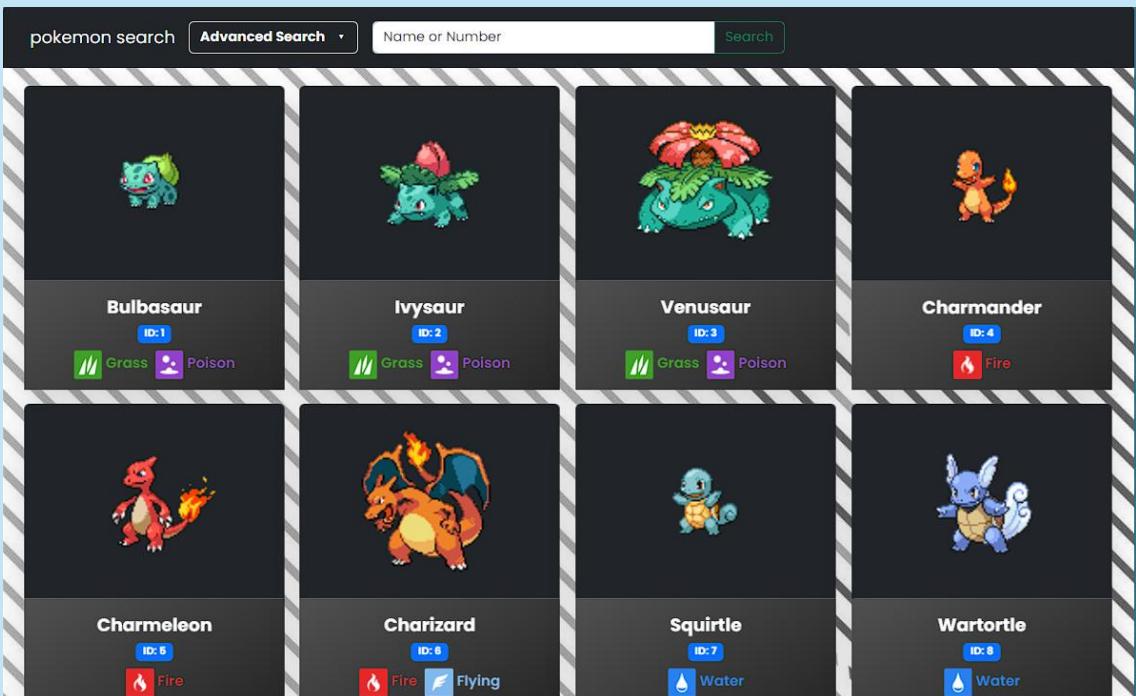
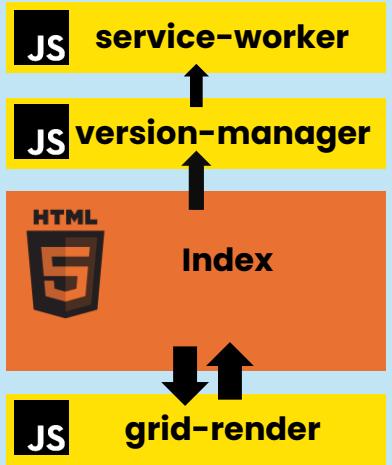
Forwards request
Return data

Dexie.js
IndexedDB

Items are rendered

Asks for 8 pokémon

PWA



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



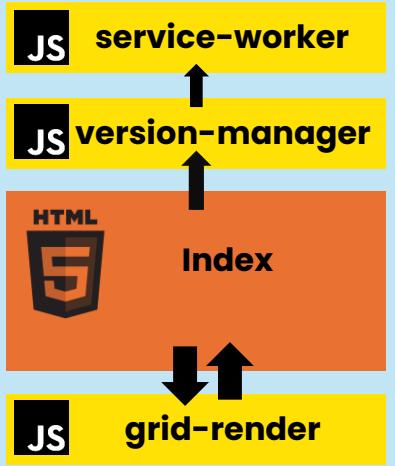
PokeAPI

Dexie.js

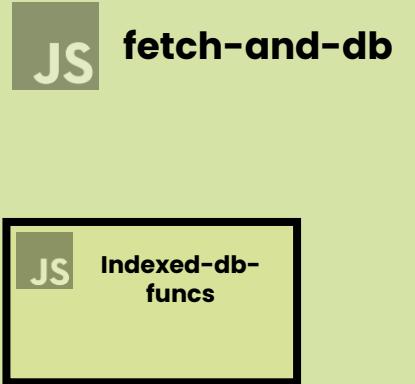
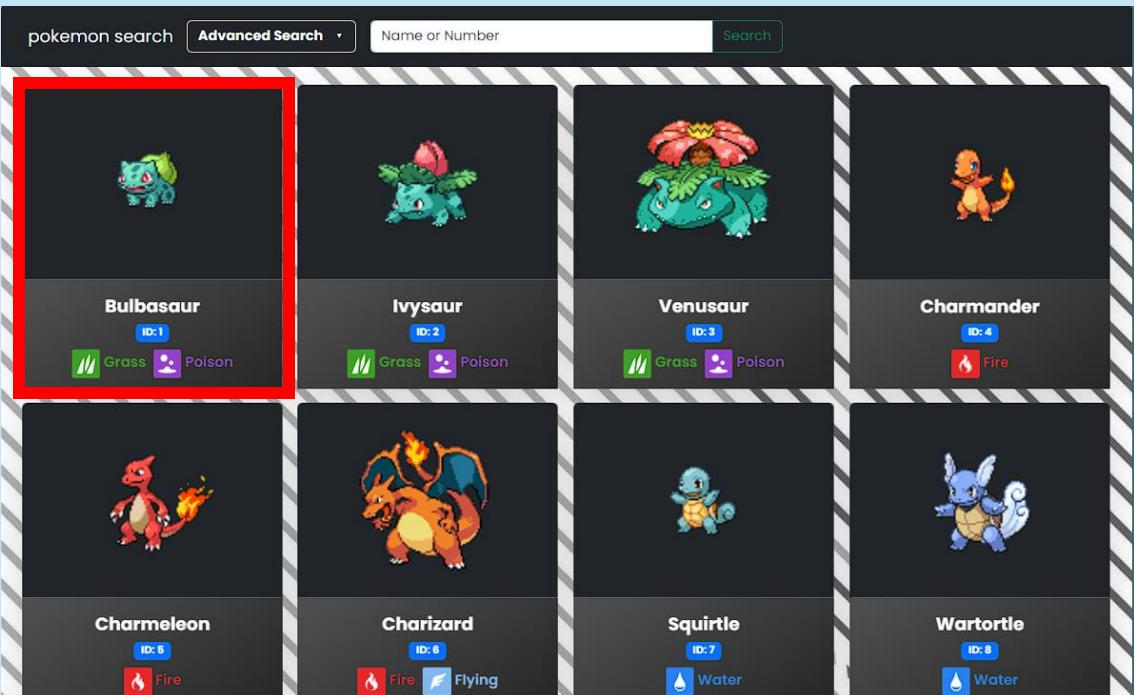
IndexedDB



PWA



If user clicks on a card...



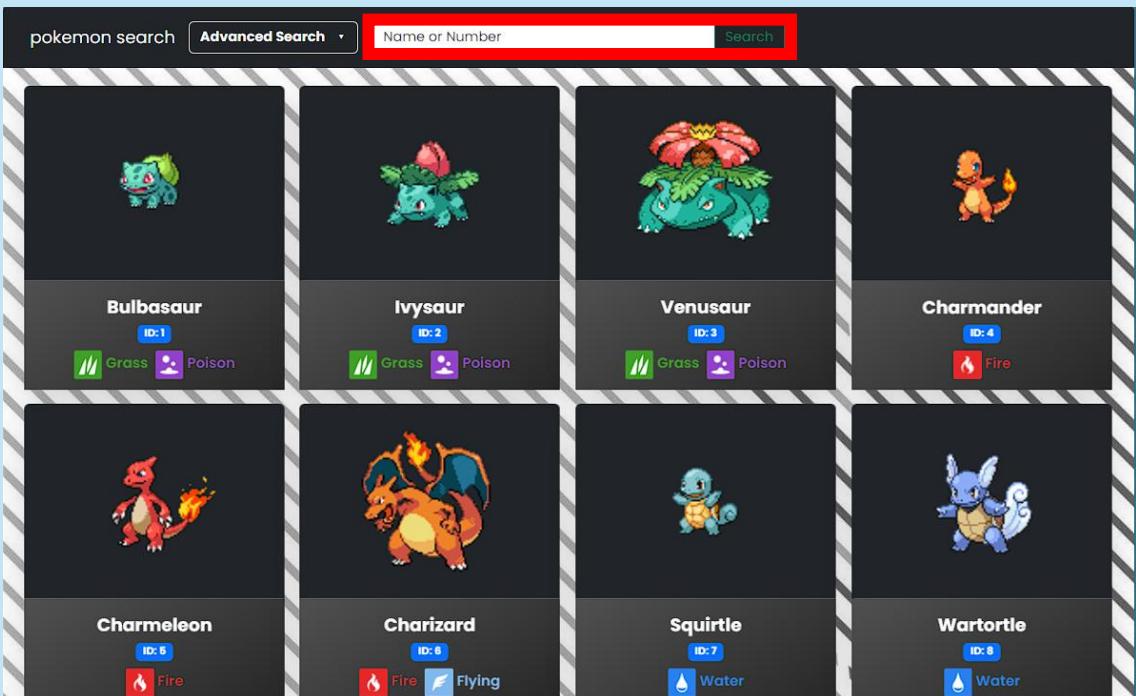
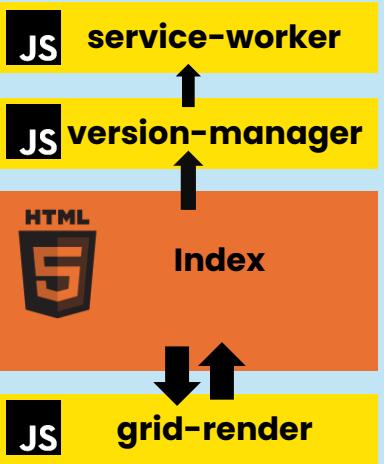
PokeAPI

Dexie.js

IndexedDB



PWA



Or if user enters an ID or Name...



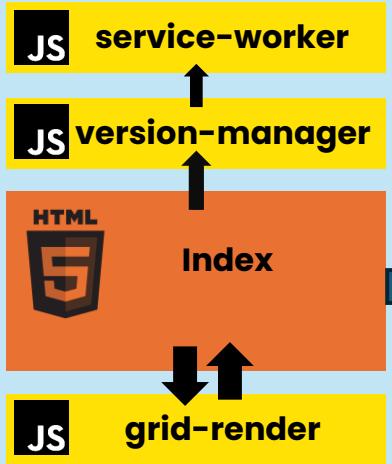
PokeAPI

Dexie.js

IndexedDB



PWA



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

```
<a href="pokemon-detail.html?id=3">
```

URL GET request

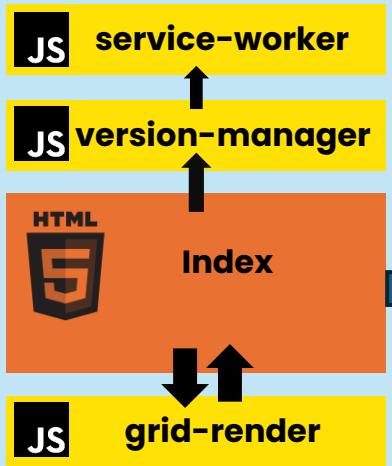


PokeAPI

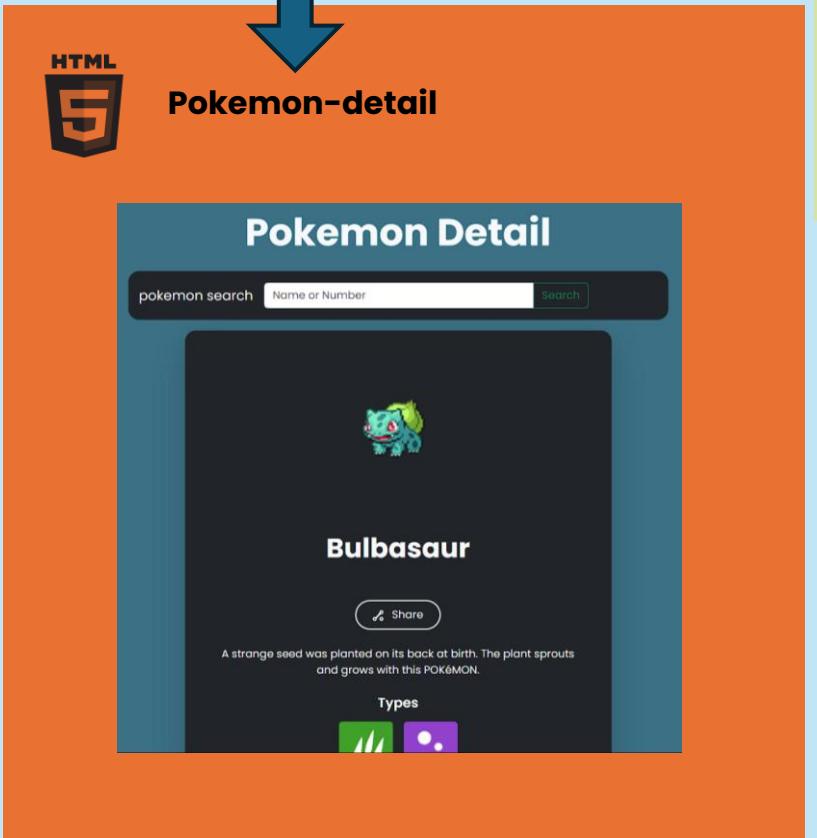
Dexie.js

IndexedDB

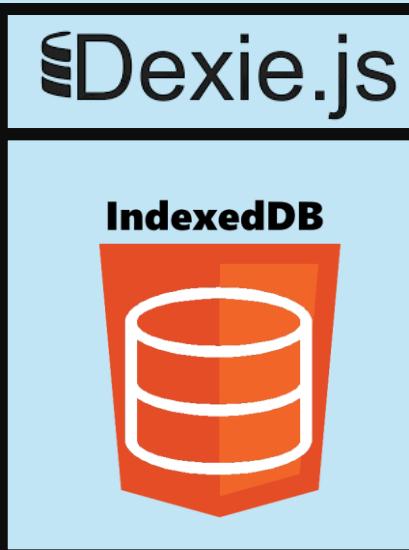


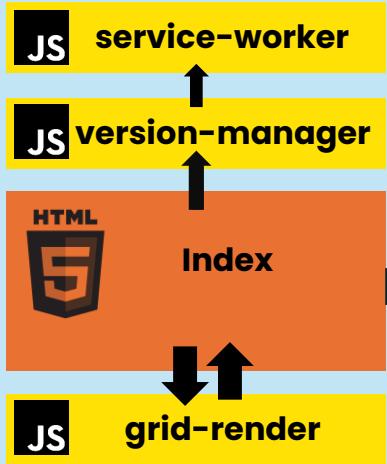


Generated link or URL GET request

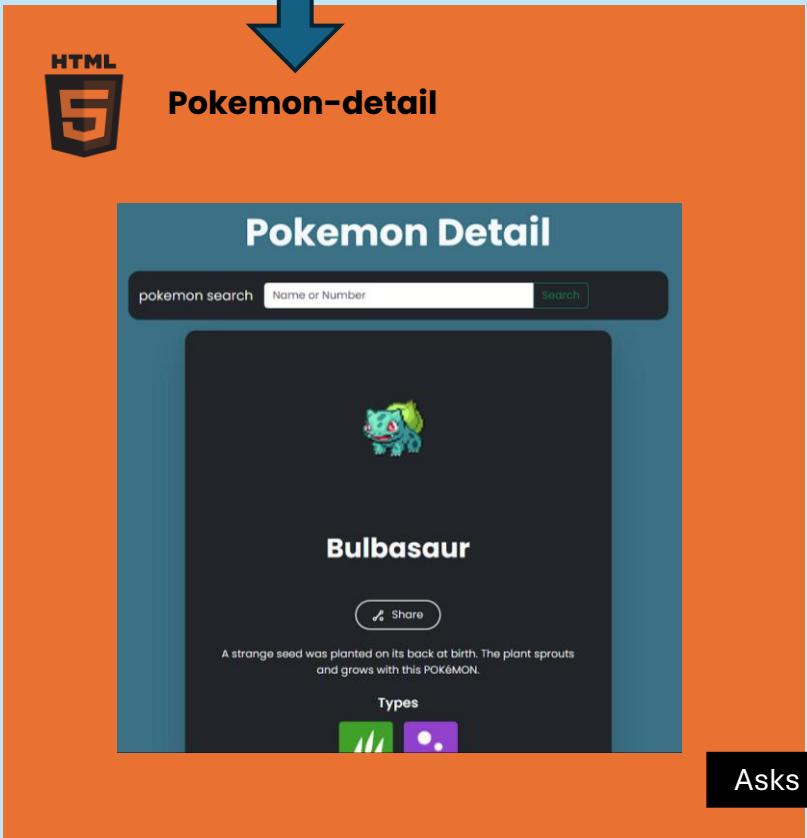


The user is sent to the page.





Generated link or URL GET request



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

JS fetch-and-db

JS Indexed-db-funcs

JS detail-render



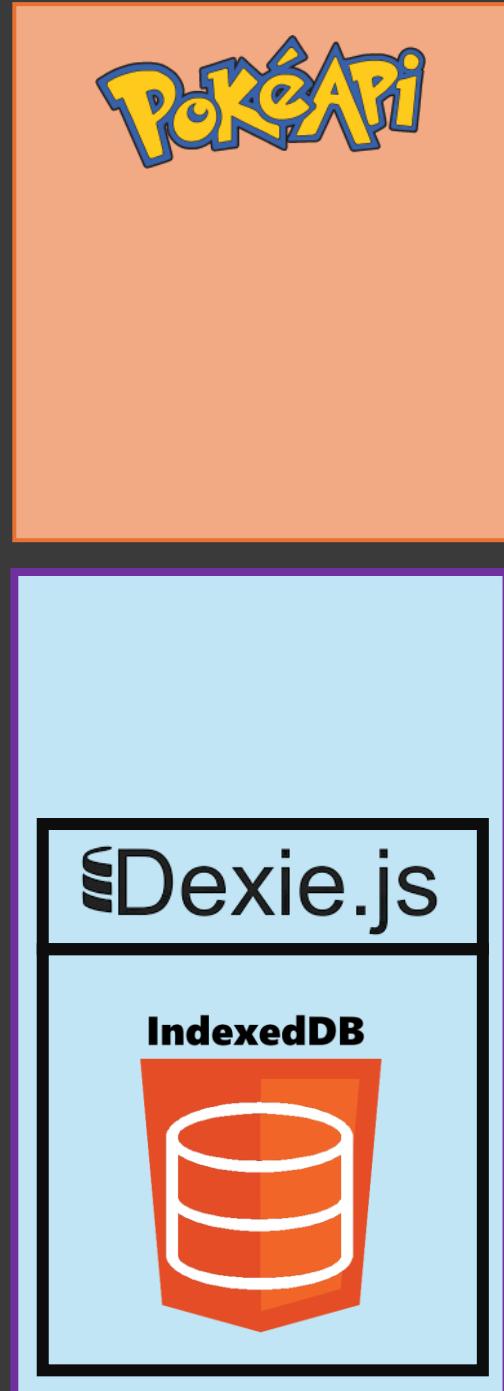
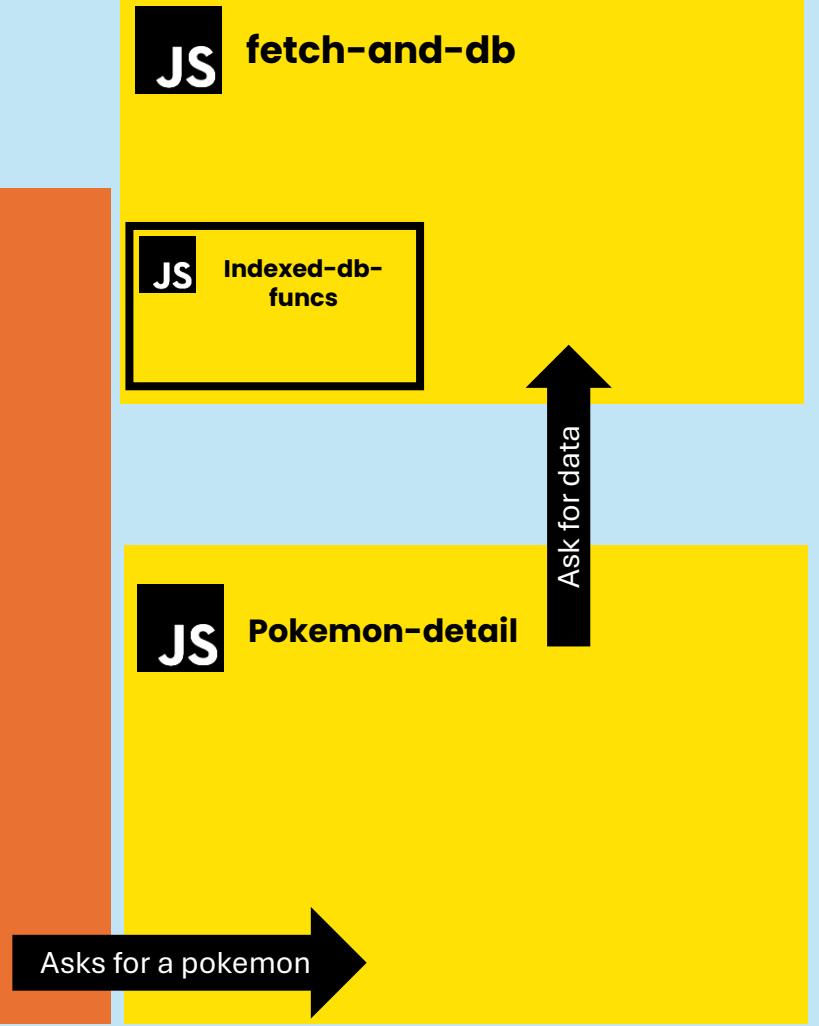
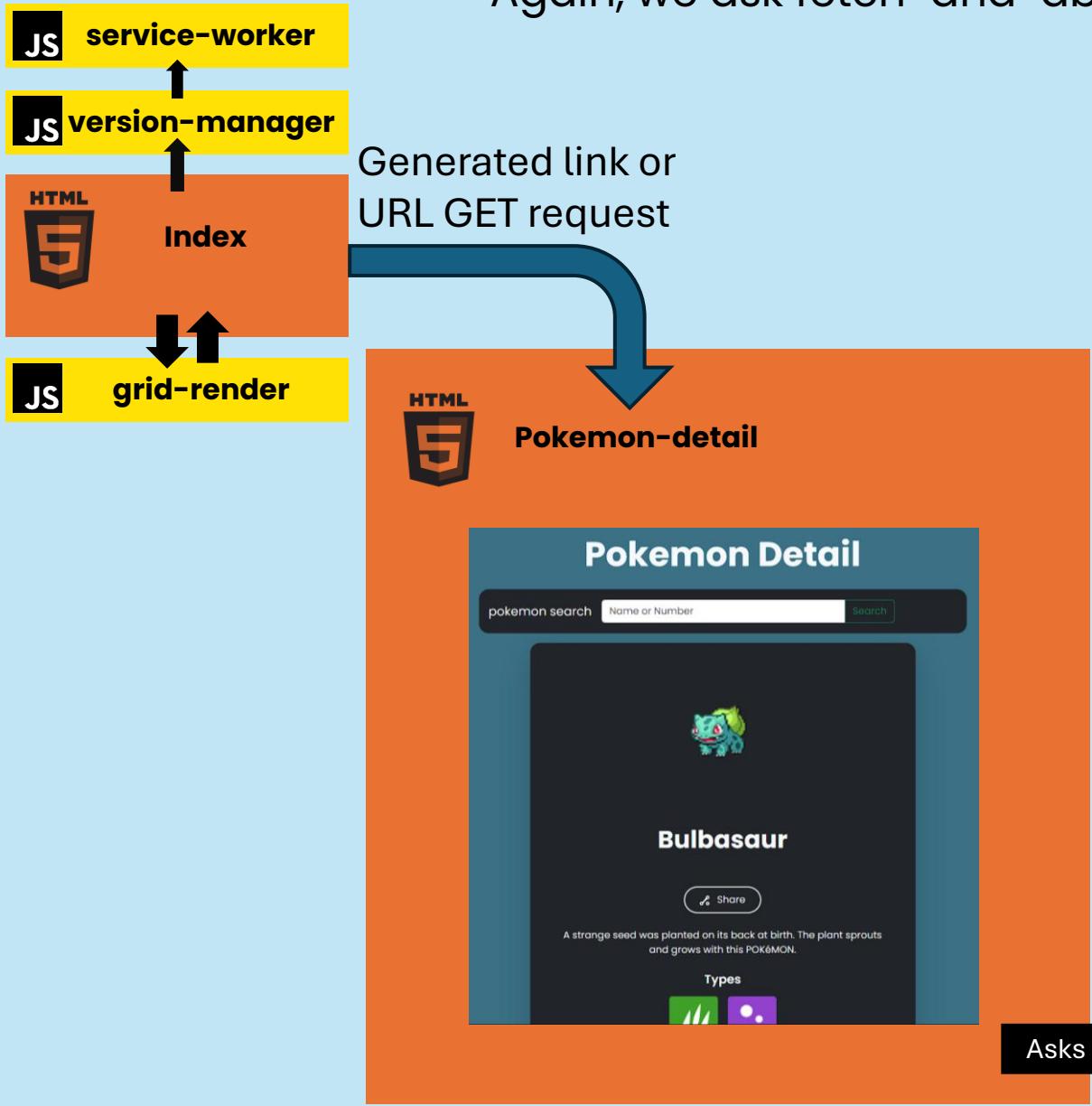
Dexie.js

IndexedDB



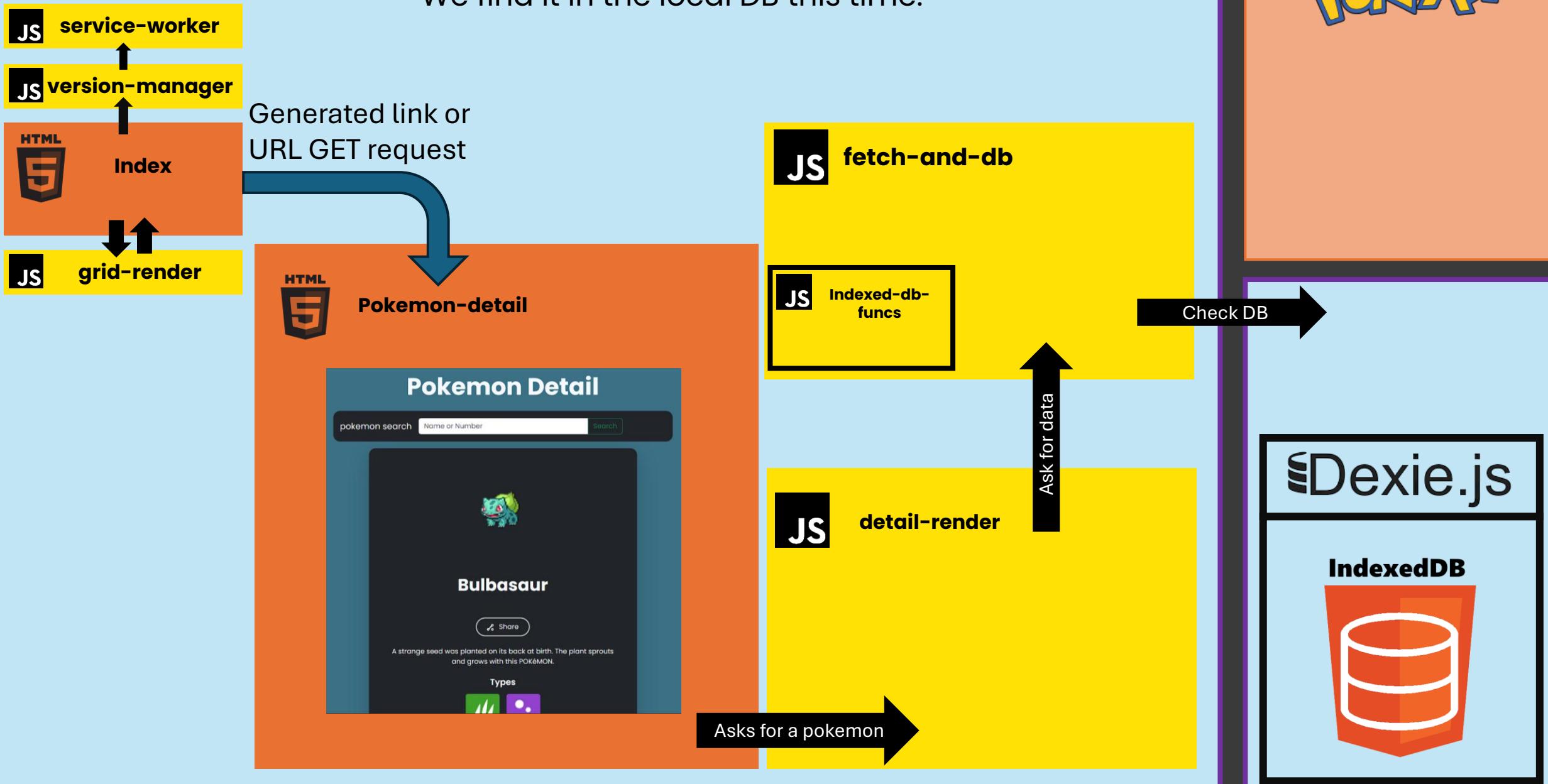


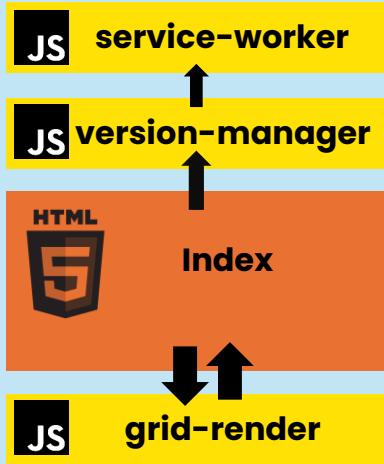
Again, we ask fetch-and-db for the json info



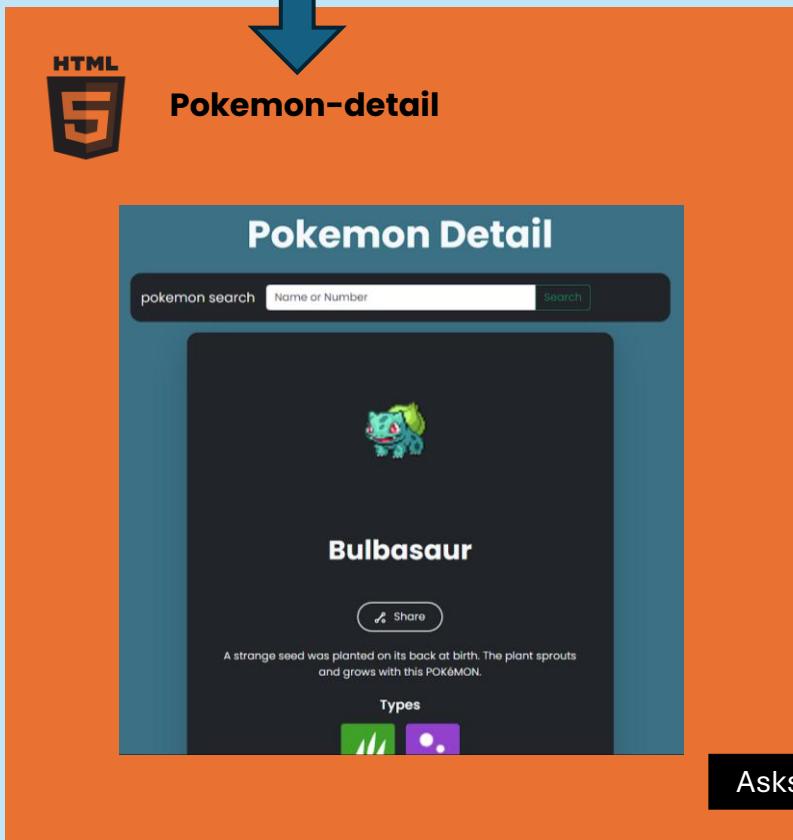


We find it in the local DB this time.



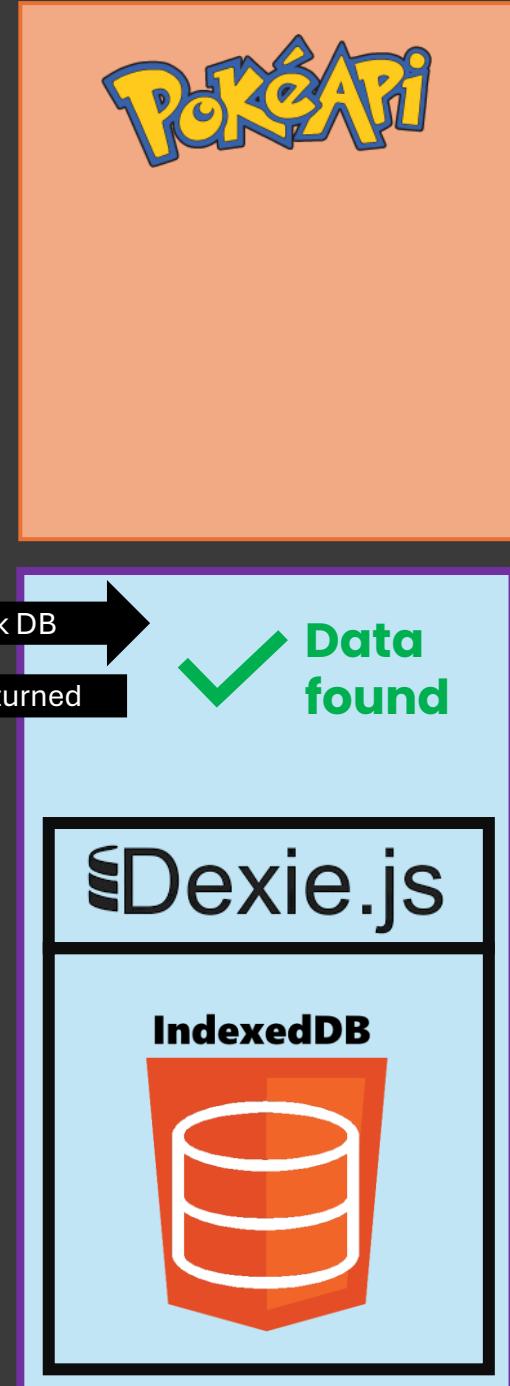
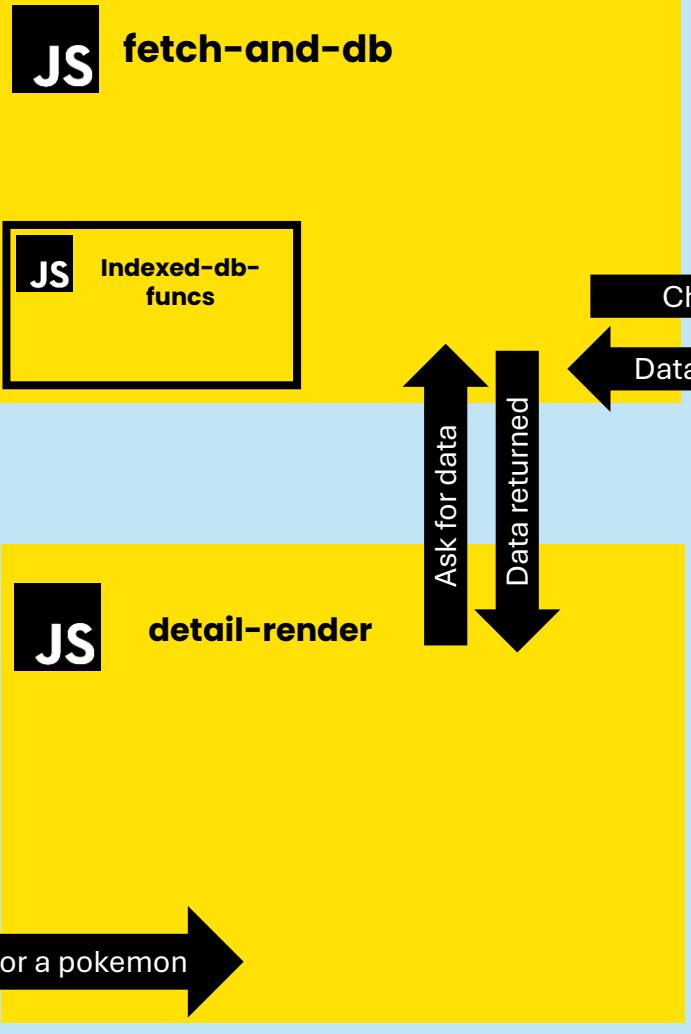


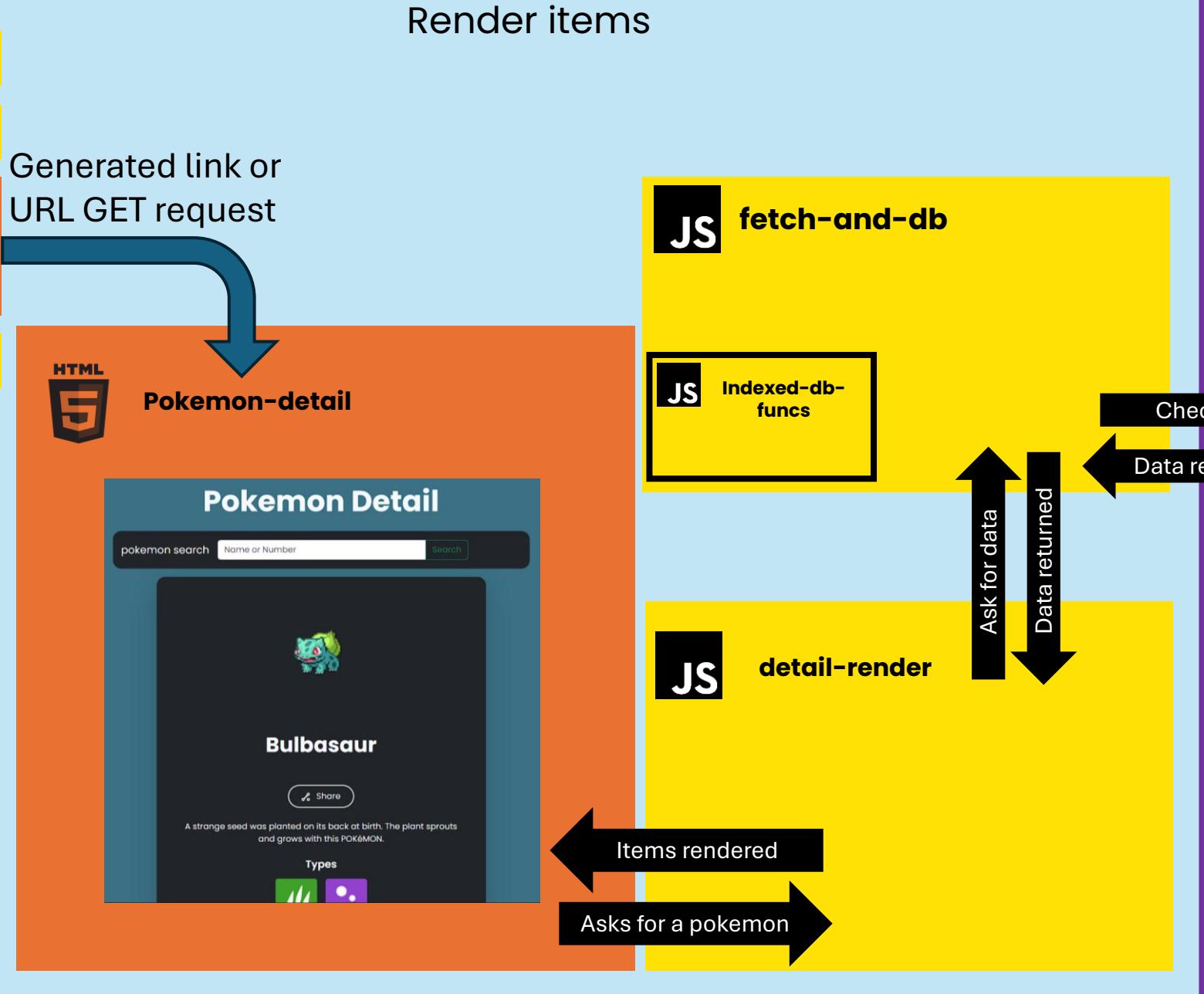
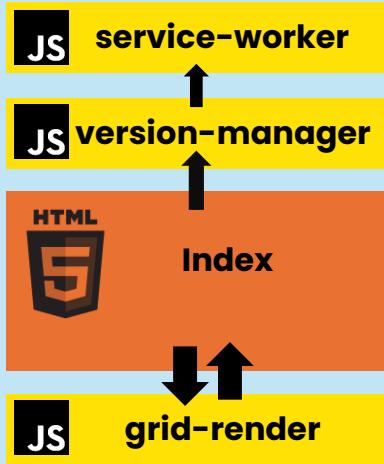
Generated link or URL GET request



Return data

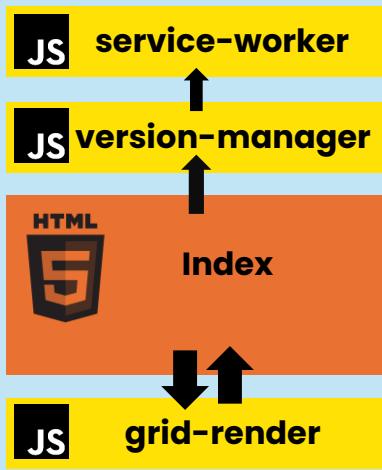
Asks for a pokemon



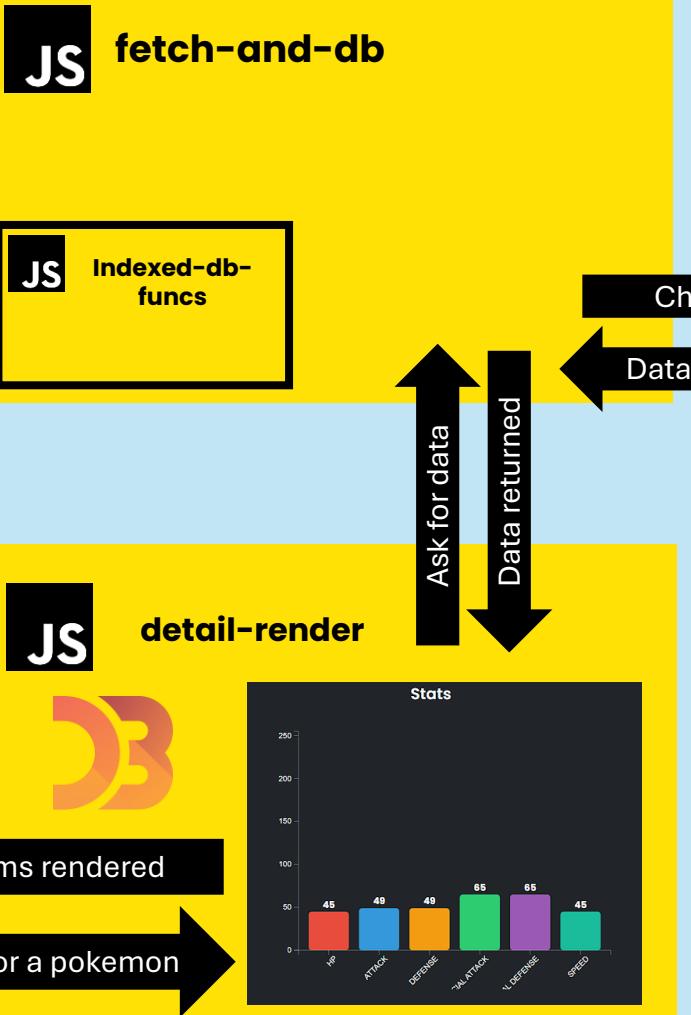
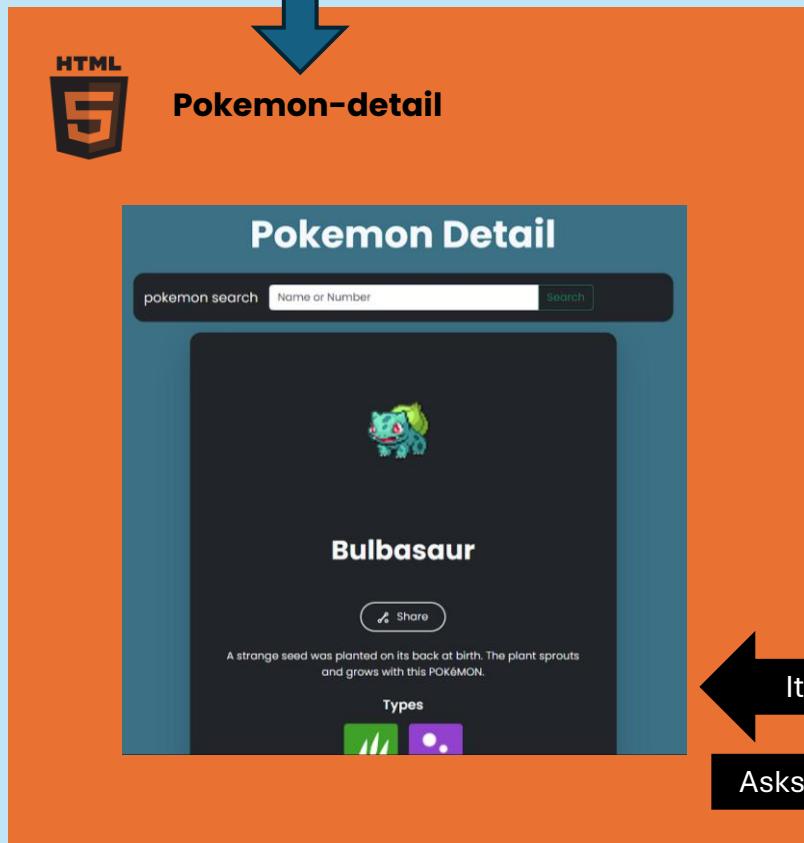




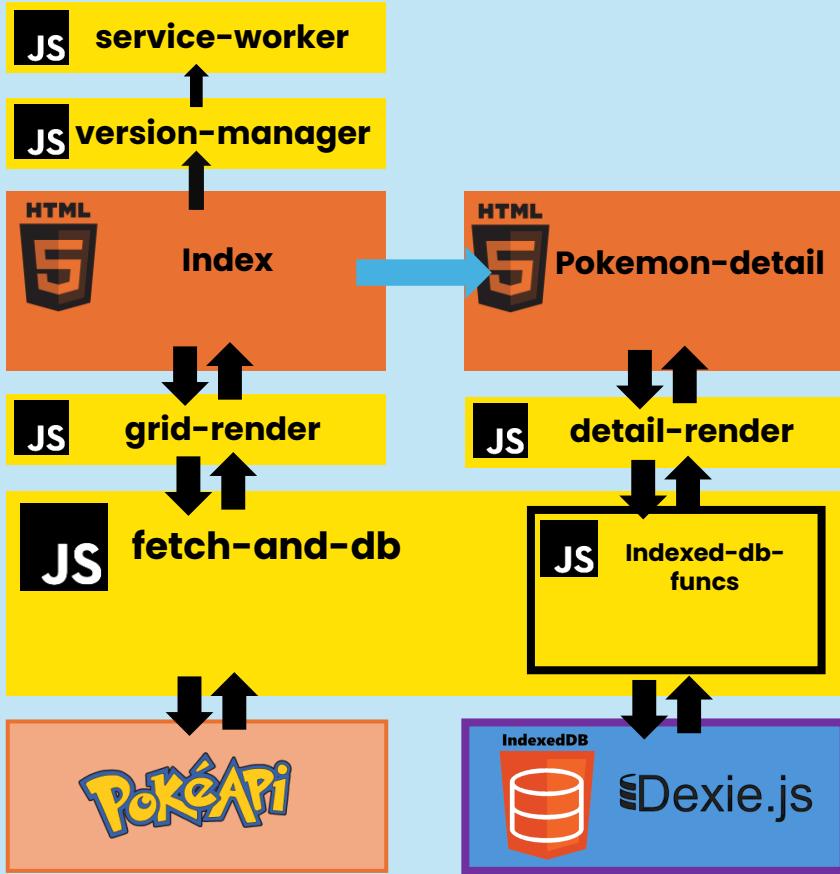
The D3 chart is created by detail-render-chart



Generated link or URL GET request



PWA



Simplified vue