Valence Labs

npm i @rdkit/rdkit

Current state, and what we can do better

Michel Moreau, Senior Software Engineer @ Valence Labs
@MichelML (GitHub)
@michelml6 (Twitter)
michelmoreau1 (LinkedIn)

Agenda

- Introduction
- Current state
- Current problems and potential solutions
- How to contribute

Valence Labs

< 2020

RDKit.js (legacy)

JavaScript wrappers around most of the RDKit. Greg reached the conclusion that it wouldn't be supportable over the long term.

2020

MinimalLib, first release

First version of a minimal, but useful, subset of RDKit functionality ported to JavaScript.

2021

NPM package, first release

Using latest docker artifacts generation features, a first version of the MinimalLib is released on npm

2021-2023

Dedicated project for rdkit-js

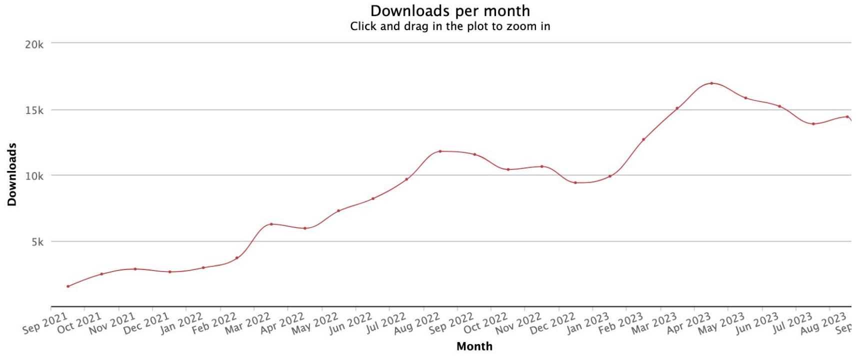
In an attempt to democratize, increase visibility, and facilitate the usage of the rdkit-js package, the rdkit-js repository has been created and is receiving open-source contributions.

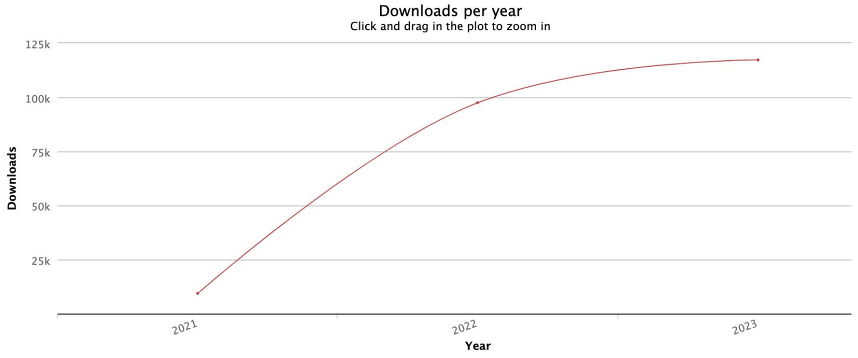
>= 2023

Where to next?

- Easier adoption (install and use)
- More off-the-shelf functionality
- Improved docs processes
- CI robust and integrated to rdkit
- Unifying efforts
- Recruiting

Current state





- 91 stars, 27 forks, used by ~50 repos
  13K downloads/month
  156k downloads/year
  208k total downloads
  ~weekly activity in repository

- Maintained, slowly evolving initiative

- **Examples in several modern frameworks**

- Dedicated GH project + Azure CD pipeline for automated package release

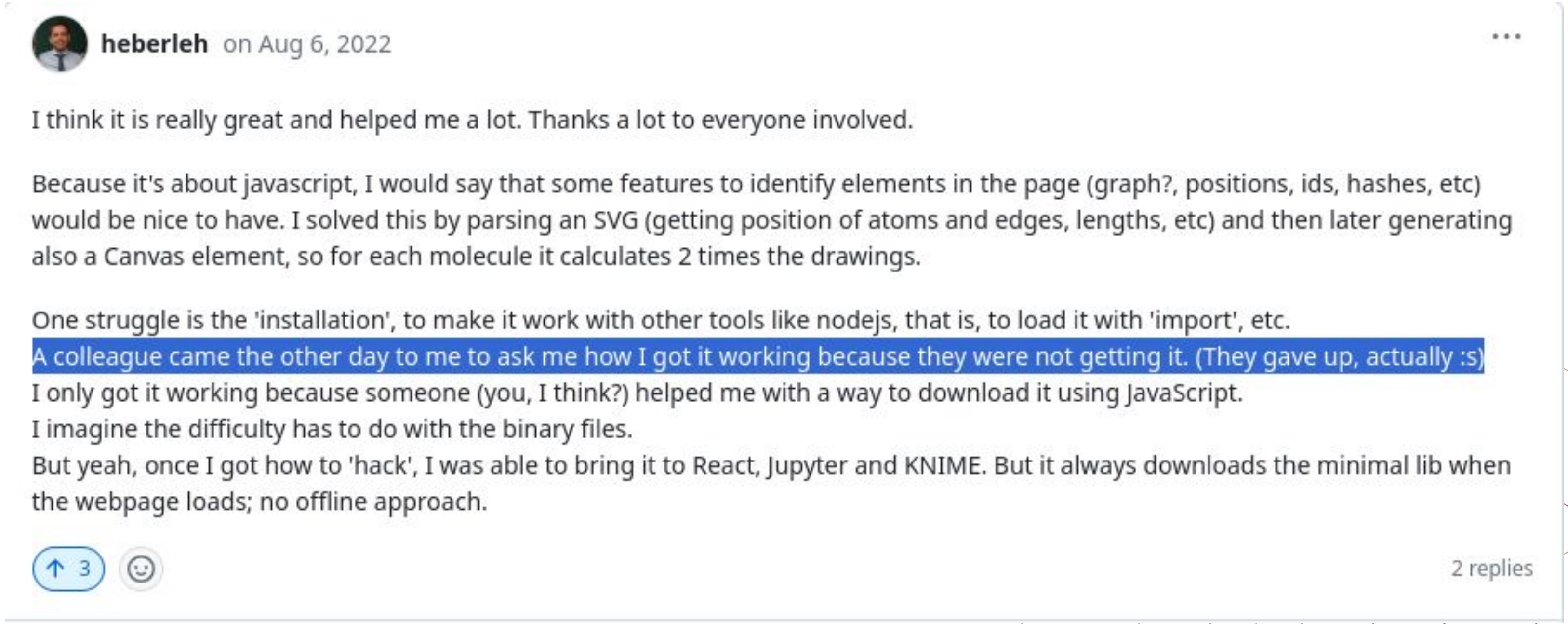
https://www.rdkitjs.com/

Valence Labs

Some problems

- Ease of adoption, difficult to set up (for first time users)
- No "real-world" off-the-shelf components for common use cases
- API docs can be tricky to maintain due to the nature of the project
- Current CI/CD process is error prone
- Focus on functionalities in different frameworks > new functionalities, approaches, and use cases.

Achieving easy adoption...



heberleh on Aug 6, 2022

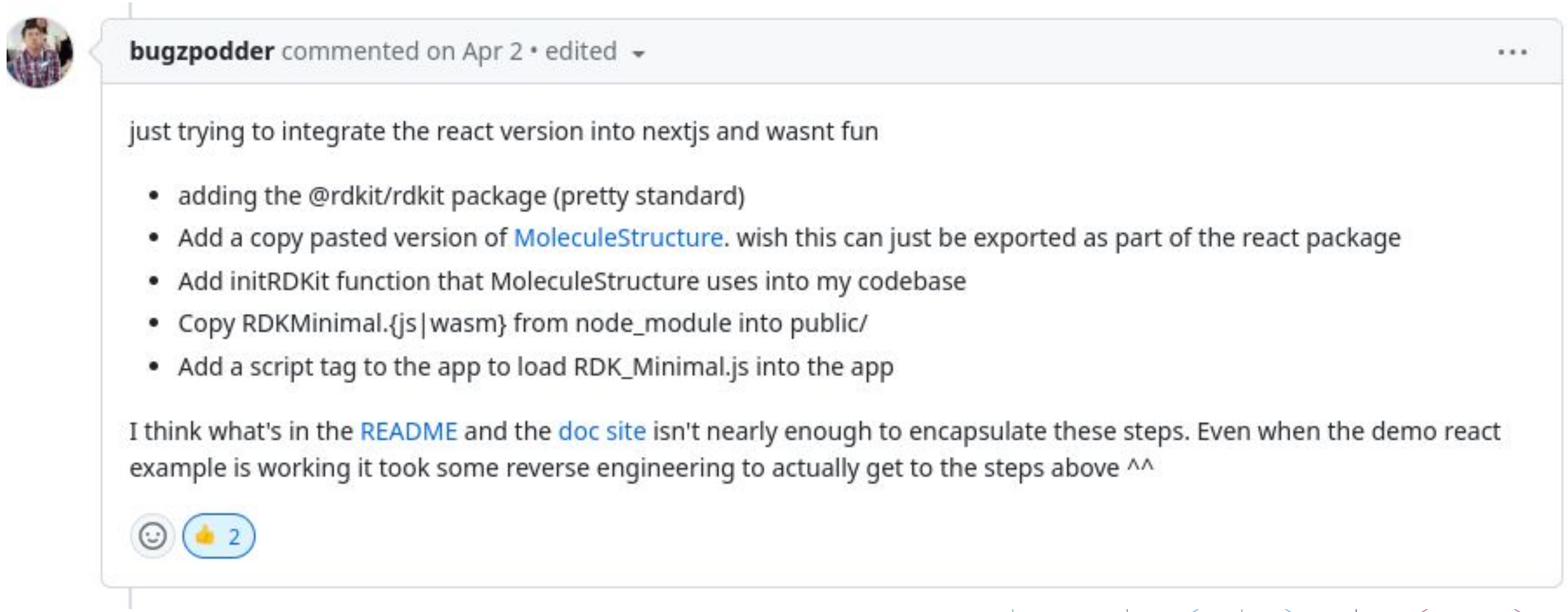
I think it is really great and helped me a lot. Thanks a lot to everyone involved.

Because it's about javascript, I would say that some features to identify elements in the page (graph?, positions, ids, hashes, etc) would be nice to have. I solved this by parsing an SVG (getting position of atoms and edges, lengths, etc) and then later generating also a Canvas element, so for each molecule it calculates 2 times the drawings.

One struggle is the 'installation', to make it work with other tools like nodejs, that is, to load it with 'import', etc.
A colleague came the other day to me to ask me how I got it working because they were not getting it. (They gave up, actually :s)
I only got it working because someone (you, I think?) helped me with a way to download it using JavaScript.
I imagine the difficulty has to do with the binary files.
But yeah, once I got how to 'hack', I was able to bring it to React, Jupyter and KNIME. But it always downloads the minimal lib when the webpage loads; no offline approach.

Valence Labs

Achieving easy adoption...



bugzpodder commented on Apr 2 • edited

just trying to integrate the react version into nextjs and wasnt fun

- adding the @rdkit/rdkit package (pretty standard)
- Add a copy pasted version of MoleculeStructure. wish this can just be exported as part of the react package
- Add initRDKit function that MoleculeStructure uses into my codebase
- Copy RDKMinimal.{js|wasm} from node_module into public/
- Add a script tag to the app to load RDK_Minimal.js into the app

I think what's in the README and the doc site isn't nearly enough to encapsulate these steps. Even when the demo react example is working it took some reverse engineering to actually get to the steps above ^^

👍 2

Valence Labs

Achieving easy adoption: Resilient initializer

Just **initRDKit()**, and get started.

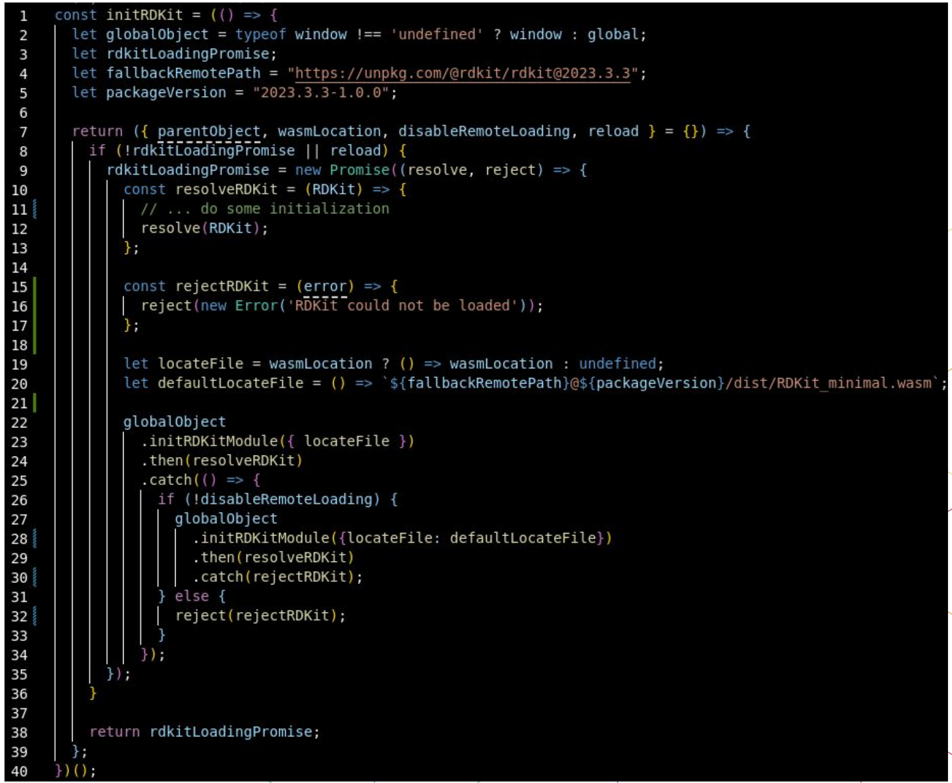
If initial **initRDKitModule()** does not work, fetch remote version from unpkg.com

Initialisation options (all optional):

- parentObject
- wasmLocation
- disableRemoteLoading
- reload

Think about WebAssembly only if you have to.

Made possible through locateFile + UNPKG .



```javascript
const initRDKit = (() => {
  let globalObject = typeof window !== 'undefined' ? window : global;
  let rdkitLoadingPromise;
  let fallbackRemotePath = "https://unpkg.com/@rdkit/rdkit@2023.3.3";
  let packageVersion = "2023.3.3-1.0.0";

  return ({ parentObject, wasmLocation, disableRemoteLoading, reload } = {}) => {
    if (!rdkitLoadingPromise || reload) {
      rdkitLoadingPromise = new Promise((resolve, reject) => {
        const resolveRDKit = (RDKit) => {
          // ... do some initialization
          resolve(RDKit);
        };

        const rejectRDKit = (error) => {
          reject(new Error('RDKit could not be loaded'));
        };

        let locateFile = wasmLocation ? () => wasmLocation : undefined;
        let defaultLocateFile = () => `${fallbackRemotePath}@${packageVersion}/dist/RDKit_minimal.wasm`;

        globalObject
          .initRDKitModule({ locateFile })
          .then(resolveRDKit)
          .catch(() => {
            if (!disableRemoteLoading) {
              globalObject
                .initRDKitModule({locateFile: defaultLocateFile})
                .then(resolveRDKit)
                .catch(rejectRDKit);
            } else {
              reject(rejectRDKit);
            }
          });
      });
    }

    return rdkitLoadingPromise;
  };
})();
```

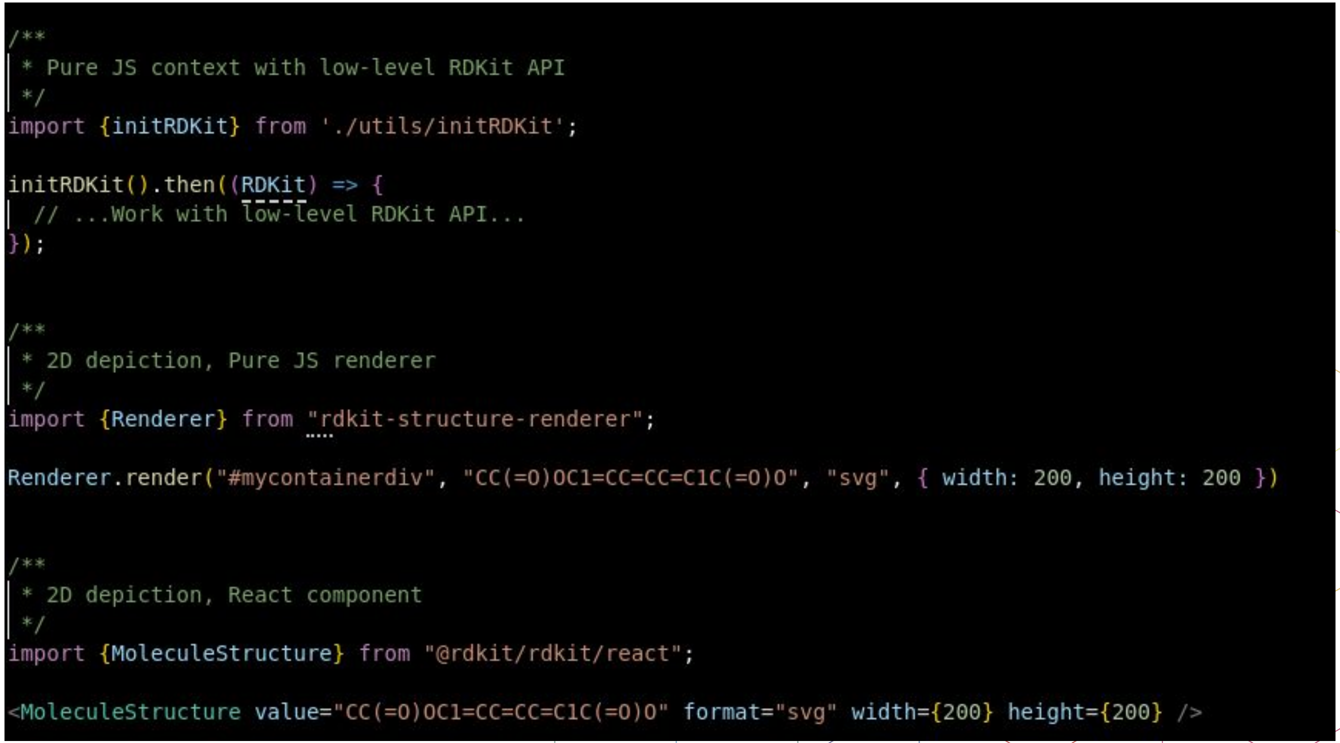
Valence Labs

Achieving easy adoption: Install, use.

A resilient initializer makes it possible to hide initialization entirely in common use cases (mostly rendering 2D depiction of molecules).

Initialization can happen on first usage of a render function or component.

For functions with a more synchronous nature, we'll always have to explicitly initialize first.



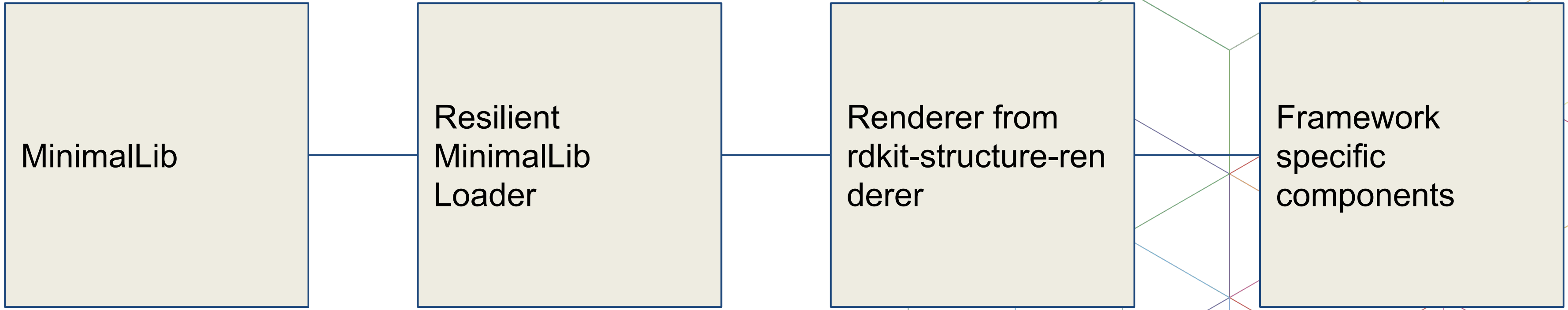
```
/**
 * Pure JS context with low-level RDKit API
 */
import {initRDKit} from './utils/initRDKit';

initRDKit().then((RDKit) => {
  // ...Work with low-level RDKit API...
});


/**
 * 2D depiction, Pure JS renderer
 */
import {Renderer} from "rdkit-structure-renderer";

Renderer.render("#mycontainerdiv", "CC(=O)OC1=CC=CC=C1C(=O)O", "svg", { width: 200, height: 200 })


/**
 * 2D depiction, React component
 */
import {MoleculeStructure} from "@rdkit/rdkit/react";

<MoleculeStructure value="CC(=O)OC1=CC=CC=C1C(=O)O" format="svg" width={200} height={200} />
```

Valence Labs

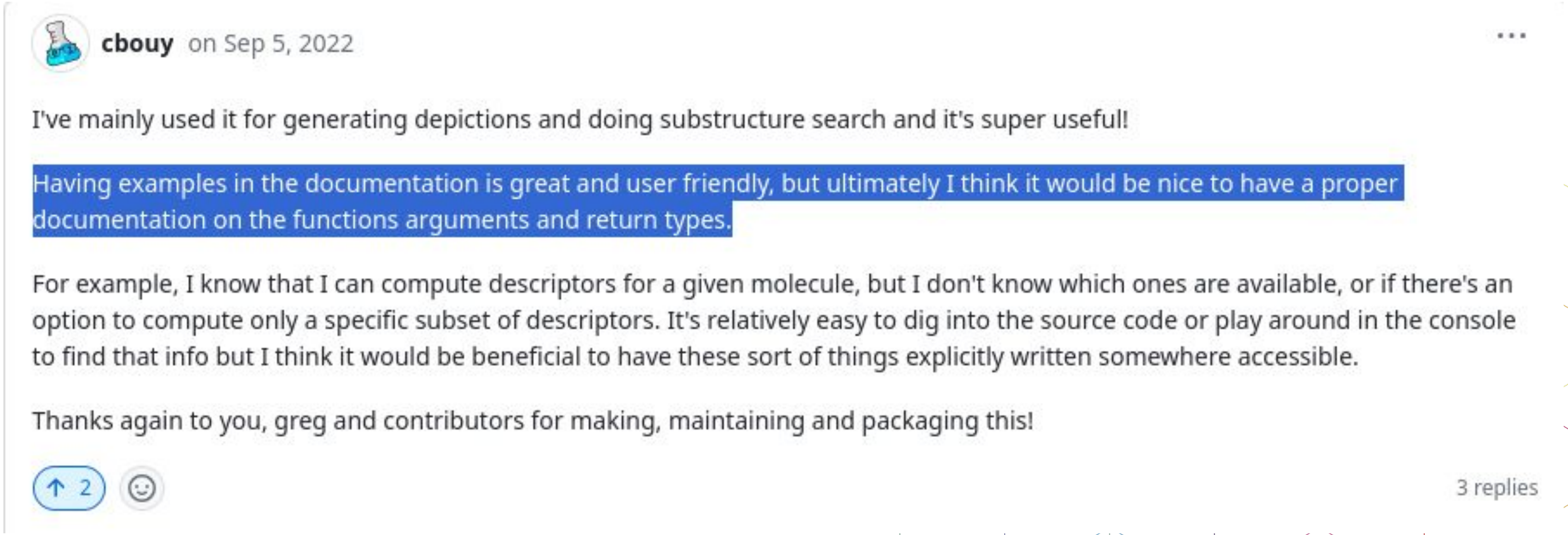
Achieving easy adoption: centralizing efforts

- The approach by Paolo Tosco (@ptosco) rdkit-structure-renderer is the de facto client-side way of leveraging the minimal lib to depict molecules in a performant manner.
    - Honorable mention to the @iktos/molecule-representation initiative

- Let's build a strong foundation on which potential framework specific components can be built


MinimalLib
Resilient MinimalLib Loader
Renderer from rdkit-structure-renderer
Framework specific components

Achieving better documentation...



cbouy on Sep 5, 2022

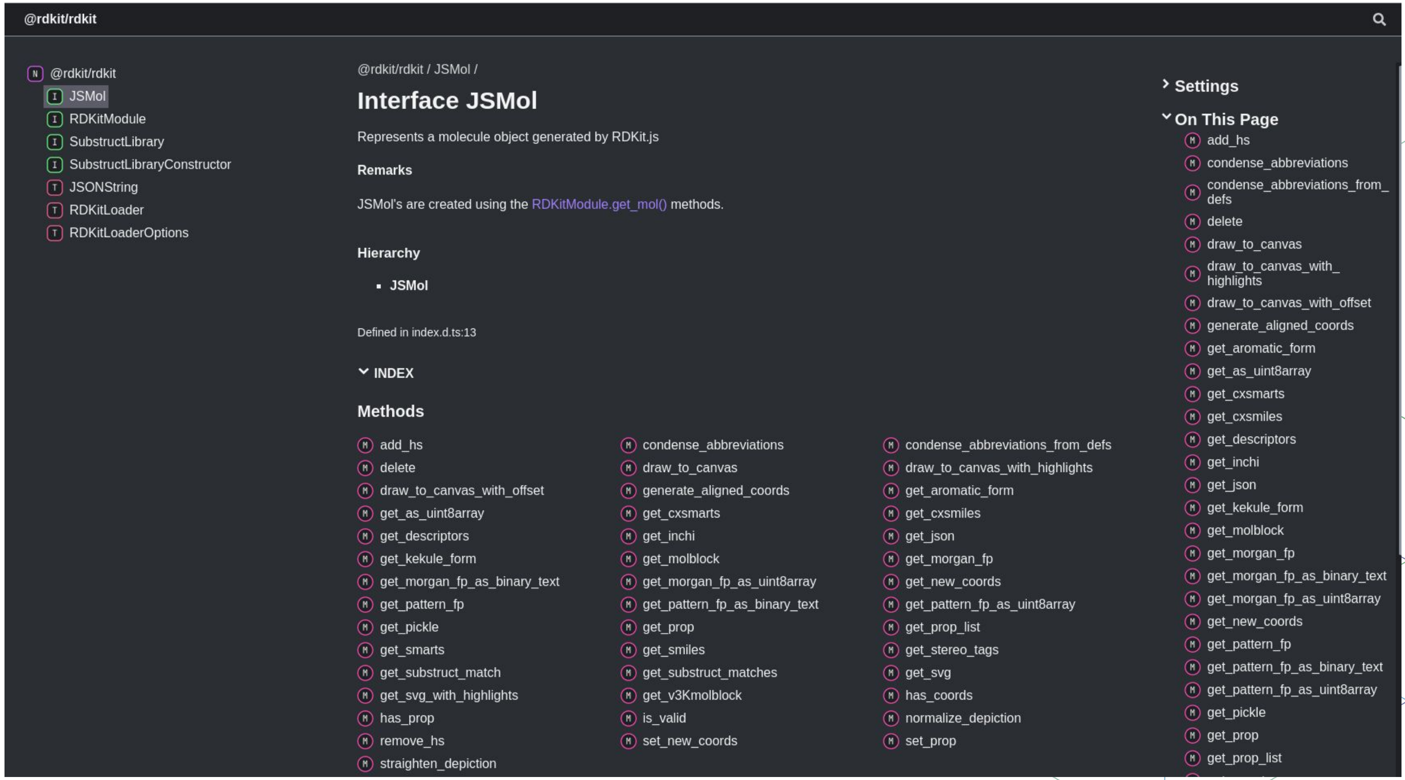
I've mainly used it for generating depictions and doing substructure search and it's super useful!

Having examples in the documentation is great and user friendly, but ultimately I think it would be nice to have a proper documentation on the functions arguments and return types.

For example, I know that I can compute descriptors for a given molecule, but I don't know which ones are available, or if there's an option to compute only a specific subset of descriptors. It's relatively easy to dig into the source code or play around in the console to find that info but I think it would be beneficial to have these sort of things explicitly written somewhere accessible.

Thanks again to you, greg and contributors for making, maintaining and packaging this!

Achieving better documentation: TypeDoc



@rdkit/rdkit

N @rdkit/rdkit
- I JSMol
- I RDKitModule
- I SubstructLibrary
- I SubstructLibraryConstructor
- T JSONString
- T RDKitLoader
- T RDKitLoaderOptions

@rdkit/rdkit / JSMol /

Interface JSMol

Represents a molecule object generated by RDKit.js

Remarks

JSMol's are created using the RDKitModule.get_mol() methods.

Hierarchy

- **JSMol**

Defined in index.d.ts:13

INDEX

Methods

add_hs	condense_abbreviations	condense_abbreviations_from_defs
delete	draw_to_canvas	draw_to_canvas_with_highlights
draw_to_canvas_with_offset	generate_aligned_coords	get_aromatic_form
get_as_uint8array	get_cxsmarts	get_cxsmiles
get_descriptors	get_inchi	get_json
get_kekule_form	get_molblock	get_morgan_fp
get_morgan_fp_as_binary_text	get_morgan_fp_as_uint8array	get_new_coords
get_pattern_fp	get_pattern_fp_as_binary_text	get_pattern_fp_as_uint8array
get_pickle	get_prop	get_prop_list
get_smarts	get_smiles	get_stereo_tags
get_substruct_match	get_substruct_matches	get_svg
get_svg_with_highlights	get_v3Kmolblock	has_coords
has_prop	is_valid	normalize_depiction
remove_hs	set_new_coords	set_prop
straighten_depiction		

Settings

On This Page
- add_hs
- condense_abbreviations
- condense_abbreviations_from_defs
- delete
- draw_to_canvas
- draw_to_canvas_with_highlights
- draw_to_canvas_with_offset
- generate_aligned_coords
- get_aromatic_form
- get_as_uint8array
- get_cxsmarts
- get_cxsmiles
- get_descriptors
- get_inchi
- get_json
- get_kekule_form
- get_molblock
- get_morgan_fp
- get_morgan_fp_as_binary_text
- get_morgan_fp_as_uint8array
- get_new_coords
- get_pattern_fp
- get_pattern_fp_as_binary_text
- get_pattern_fp_as_uint8array
- get_pickle
- get_prop
- get_prop_list

Adam Baroti (@adam-of-barrot) wrote the initial typescript definition files of the library by hand.

And this allowed us to deploy a website describing the full MinimalLib API based on the TS definitions with the TypeDoc lib.

But, maintaining this by hand is error-prone. This is not 100% up to date at the moment.

Valence Labs

Hackathon idea: Use an LLM chain to assist API documentation (.d.ts definitions) generation

- Based on very very early experiment: https://chat.openai.com/share/0d1920ed-08d0-4f35-b5f4-0530bcedfff2

- Can we achieve something that helps with LLMs by:
  - Providing the MinimalLib source code
  - Providing the main RDKit release notes
  - Creating the whole index.d.ts generation using smaller tasks
  - Asking the LLM to adapt the generation based on the existing index.d.ts
  - Making this process a pre-release step in the CI/CD pipeline of rdkit-js package release.

Valence Labs

Other problems/solutions

- Problem: CI/CD is error-prone because it is not run as part of the main rdkit project CI.
  Solution:
  → Adapt the rdkit-js CD pipeline to a CI pipeline integrated into rdkit.

- Problem: Showcasing the same functionalities many times > new functionalities.
  Solution:
  → Log and address GH issues related to new functionalities, novel approaches, or use cases requested by the community (or entirely novel).
  → Centralization of efforts
  → Develop examples for existing features not showcased (reaction depiction, ...)
  → Move framework examples at a more discrete location

Valence Labs

What can you do to help

- Become an active maintainer
- Contribute
  - Code
  - GitHub issues
  - GitHub discussion forum
- Learn more at https://github.com/rdkit/rdkit-js/

Valence Labs

Come work with us

valencelabs.com/careers
recursion.com/careers

Michel Moreau, Senior Software Engineer @ Valence Labs
@MichelML (GitHub)
@michelml6 (Twitter)
michelmoreau1 (LinkedIn)

Thank you.

Valence Labs