



**Lucia**



# Lucia

JavaScript Library for Developing  
Performance-Focused Web  
Applications

Aiden Bai

# Engineering Problem & Goal

- **Old Websites**

- Simple and static
- Non-interactive

- **New Websites**

- Interactive
- Slow and complex
- Bad user & developer experience

- Major issue for web developers

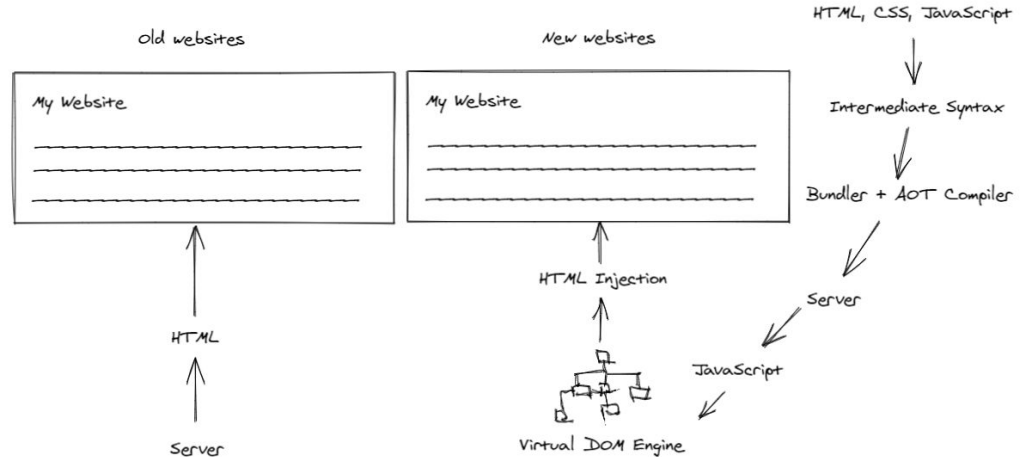


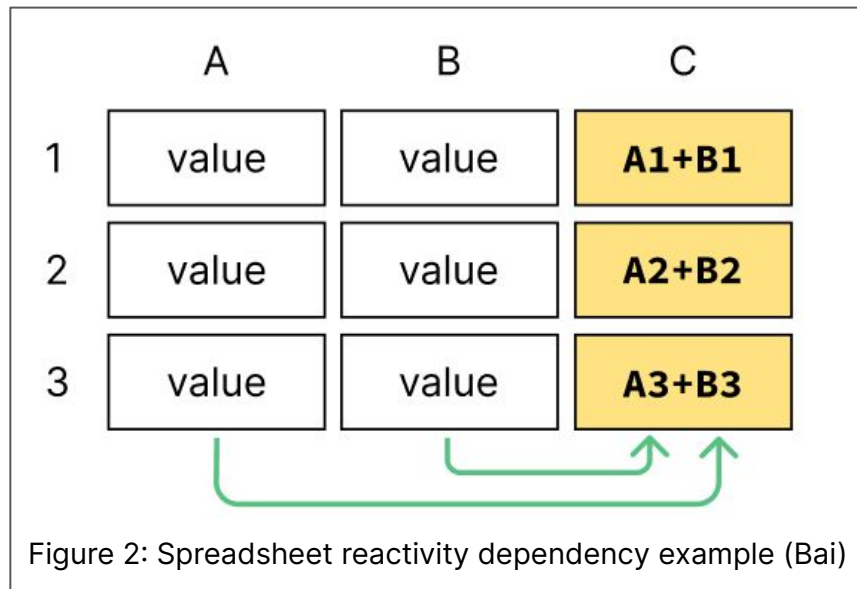
Figure 1: Old vs. New websites diagram (Bai)



**Lucia**

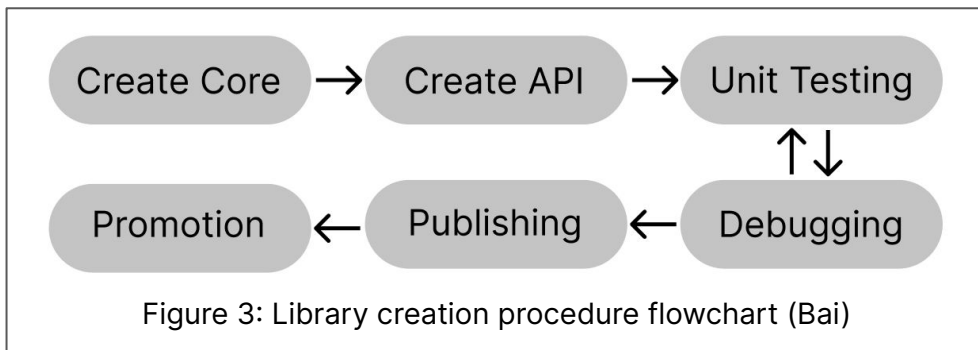
# Design Principles

- **Speed**
  - Conservative execution
  - Immutable patterns
  - Lightweight core
- **Simplicity**
  - Scalability
  - Reactivity



# Methods & Procedure

- **Language:** TypeScript
- **Build:** Rollup/Babel
- **Repository:** Github
- **Publish:** NPM + Unpkg
- **Testing:** Jest, Ad-hoc
- **Code Editor:** VSCode



# Implementation

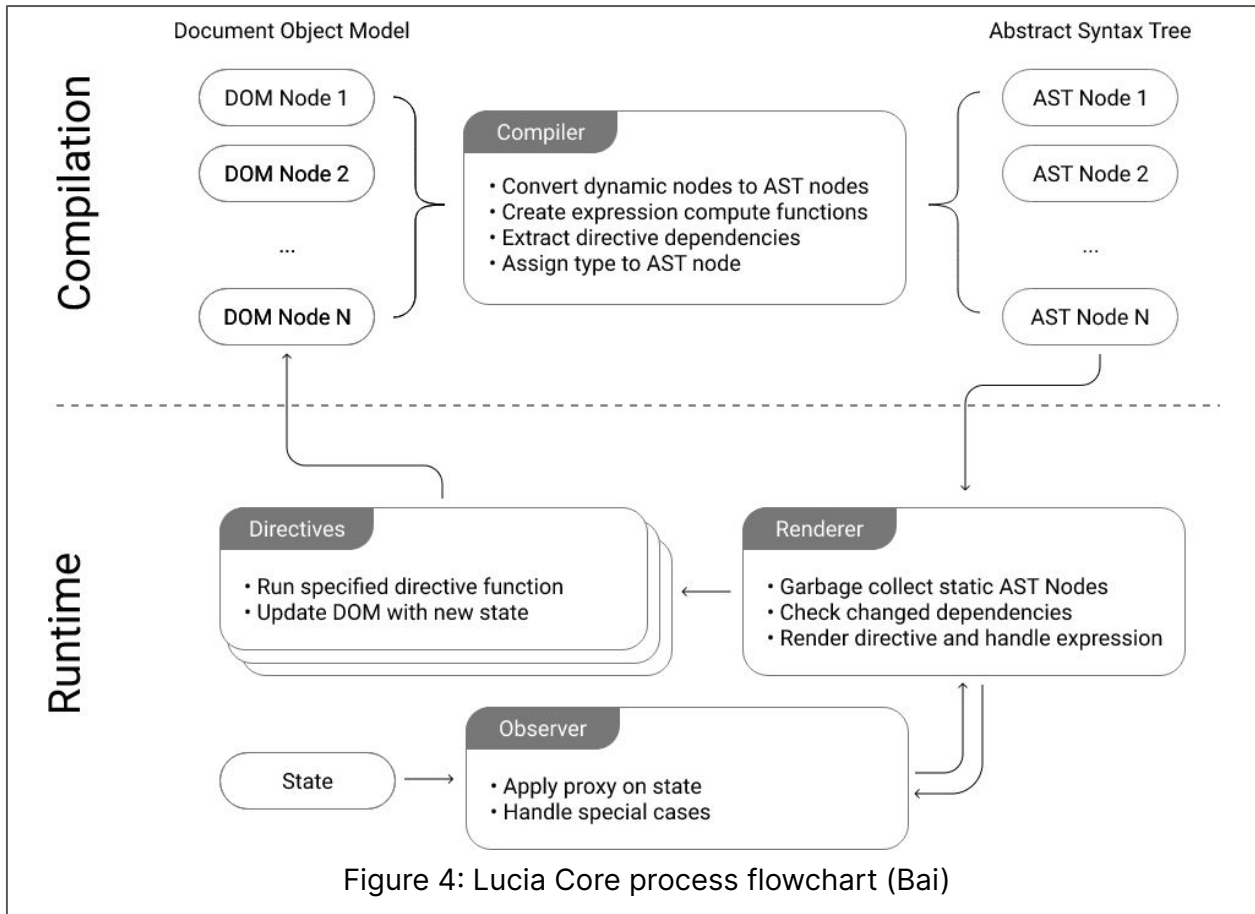


Figure 4: Lucia Core process flowchart (Bai)

# Results & Analysis

Library	None	Lucia	React 17	Angular 9	Vue 3	jQuery
Raw size (kb)	0.00	2.29	156.00	108.00	342.00	409.00
Minified+Brotli Size (kb)	0.00	1.87	19.80	65.40	94.50	86.30
Script bootup time Non-keyed (ms)	16.0	16.0	16.0	114.8	16.0	69.1
Create 1000 Rows Non-Keyed (ms)	107.2	108.1	183.5	152.1	152.1	209.9
Update 1000 Rows Non-Keyed (ms)	35.4	36.6	45.0	39.6	45.8	205.2
Clear 1000 Rows Non-Keyed (ms)	101.6	102.5	147.6	245.1	158.7	217.5
Geometric Mean of Benchmarks	49.8	50.5	66.5	114.1	64.9	159.5
Mean of Sizes	0.00	2.08	87.90	86.70	218.25	247.65
Weighted Score	24.92	26.28	77.18	100.40	141.55	203.58
Learning Curve	Low	Low	Medium	High	Medium	Low
DOM Type	DOM	DOM	VDOM	DOM	VDOM	DOM
Abstraction	None	Weak	Weak	Strong	Weak	Weak
Rendering	Client	Client	Client/SSR	Client/SSR	Client/SRR	Client

Figure 5: Comparison table of major JavaScript Libraries (Bai)

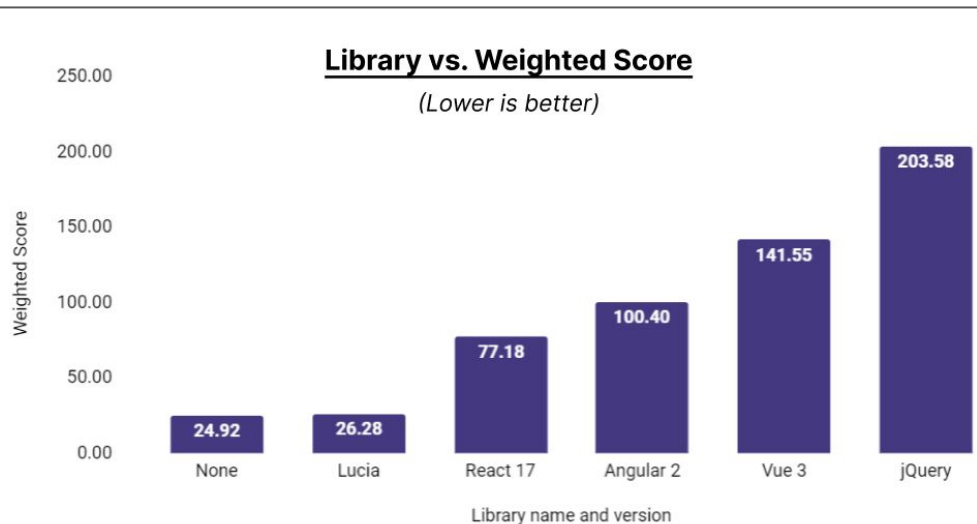


Figure 6: Column Graph of weighted score for JavaScript Libraries (Bai)



# Conclusion & Next Steps

- Simple to integrate and super fast
- Implemented in many use cases - production ready
- Learned about JavaScript library internals - 8 months of work
- Innovated upon pitfalls for a more useful product
- **Next Steps:**
  - Promote to more developers
  - Create tooling for 1.0.0

# References

- Jensen, S. H., Madsen, M., & Møller, A. (2011, June 11). Modeling the HTML DOM and Browser API in Static Analysis of JavaScript Web Applications [Modeling the HTML DOM and Browser API in Static Analysis of JavaScript Web Applications]. 19th ACM SIGSOFT symposium. Retrieved October 11, 2020, from <https://users-cs.au.dk/amoeller/papers/dom/paper.pdf>
- Mariano, C. L., B.E. Hons. (2017, January). Benchmarking JavaScript Frameworks [Benchmarking JavaScript Frameworks]. Technological University Dublin. Retrieved October 10, 2020, from <https://arrow.tudublin.ie/cgi/viewcontent.cgi?article=1100&context=scschcomdis>
- Molin, E. (2016, October 16). Comparison of Single-Page Application Frameworks [Comparison of Single-Page Application Frameworks]. Diva Portal. Retrieved October 11, 2020, from <https://www.diva-portal.org/smash/get/diva2:1037481/FULLTEXT01.pdf>
- Muyldermans, D. (2019, May 10). How Does the virtual DOM compare to other DOM updating mechanisms in JavaScript frameworks? [How Does the virtual DOM compare to other DOM updating mechanisms in JavaScript frameworks?]. University of Dublin. Retrieved October 10, 2020, from <http://www.daisyms.com/THESIS.pdf>
- Persson, M. (2020, May 28). JavaScript DOM Manipulation Performance [JavaScript DOM Manipulation Performance]. Diva Portal. Retrieved October 10, 2020, from <https://www.diva-portal.org/smash/get/diva2:1436661/FULLTEXT01.pdf>