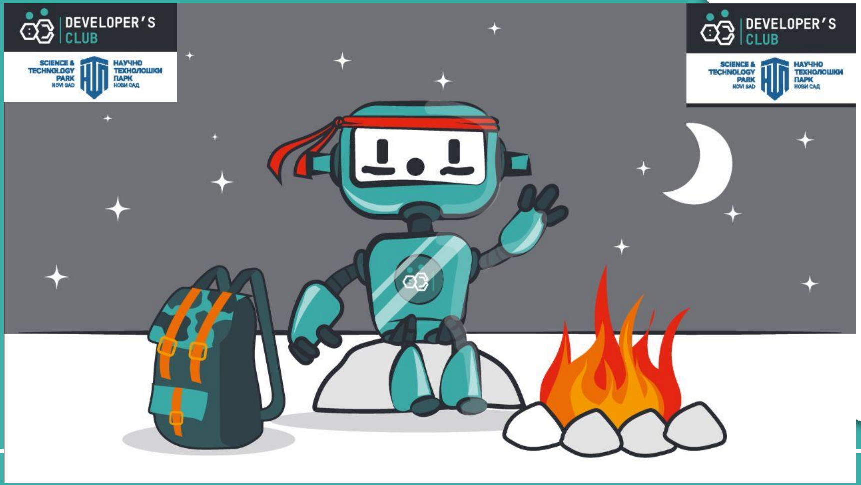
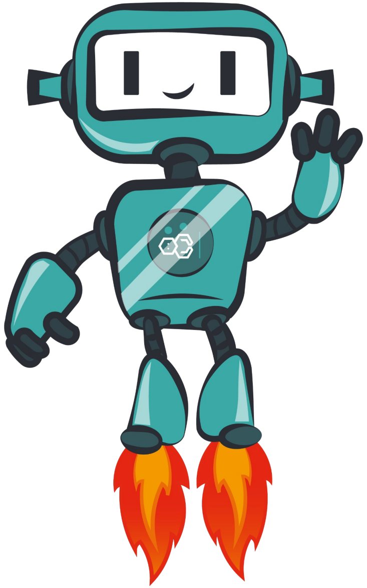
DEVELOPER'S CLUB

BDD#18 https://bit.ly/bdd-18

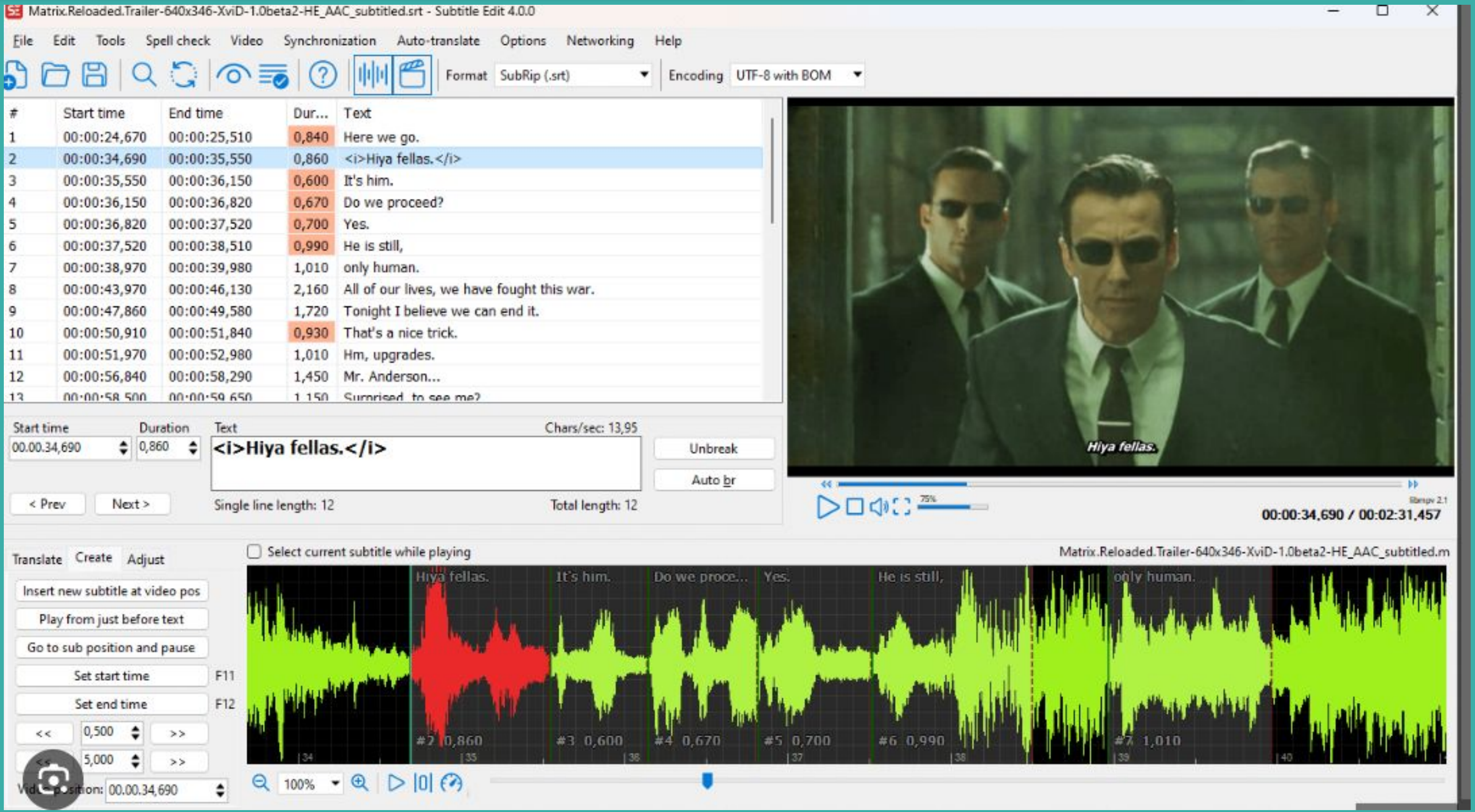
Developer War Stories!


DEVELOPER'S CLUB
SCIENCE & TECHNOLOGY PARK NOVI SAD
НАУЧНО ТЕХНОЛОШКИ ПАРК НОВИ САД
DEVELOPER'S CLUB
SCIENCE & TECHNOLOGY PARK NOVI SAD
НАУЧНО ТЕХНОЛОШКИ ПАРК НОВИ САД



Subtitles Editor

DEVELOPER'S CLUB



Matrix.Reloaded.Trailer-640x346-XviD-1.0beta2-HE_AAC_subtitled.srt - Subtitle Edit 4.0.0

Format SubRip (.srt)

Encoding UTF-8 with BOM

#	Start time	End time	Dur...	Text
1	00:00:24,670	00:00:25,510	0,840	Here we go.
2	00:00:34,690	00:00:35,550	0,860	<i>Hiya fellas.</i>
3	00:00:35,550	00:00:36,150	0,600	It's him.
4	00:00:36,150	00:00:36,820	0,670	Do we proceed?
5	00:00:36,820	00:00:37,520	0,700	Yes.
6	00:00:37,520	00:00:38,510	0,990	He is still,
7	00:00:38,970	00:00:39,980	1,010	only human.
8	00:00:43,970	00:00:46,130	2,160	All of our lives, we have fought this war.
9	00:00:47,860	00:00:49,580	1,720	Tonight I believe we can end it.
10	00:00:50,910	00:00:51,840	0,930	That's a nice trick.
11	00:00:51,970	00:00:52,980	1,010	Hm, upgrades.
12	00:00:56,840	00:00:58,290	1,450	Mr. Anderson...
13	00:00:58,500	00:00:59,650	1,150	Surprised to see me?

Start time: 00.00.34,690

Duration: 0,860

Text: <i>Hiya fellas.</i>

Chars/sec: 13,95

Single line length: 12

Total length: 12

00:00:34,690 / 00:02:31,457

☐ Select current subtitle while playing

Matrix.Reloaded.Trailer-640x346-XviD-1.0beta2-HE_AAC_subtitled.m

Set start time F11

Set end time F12

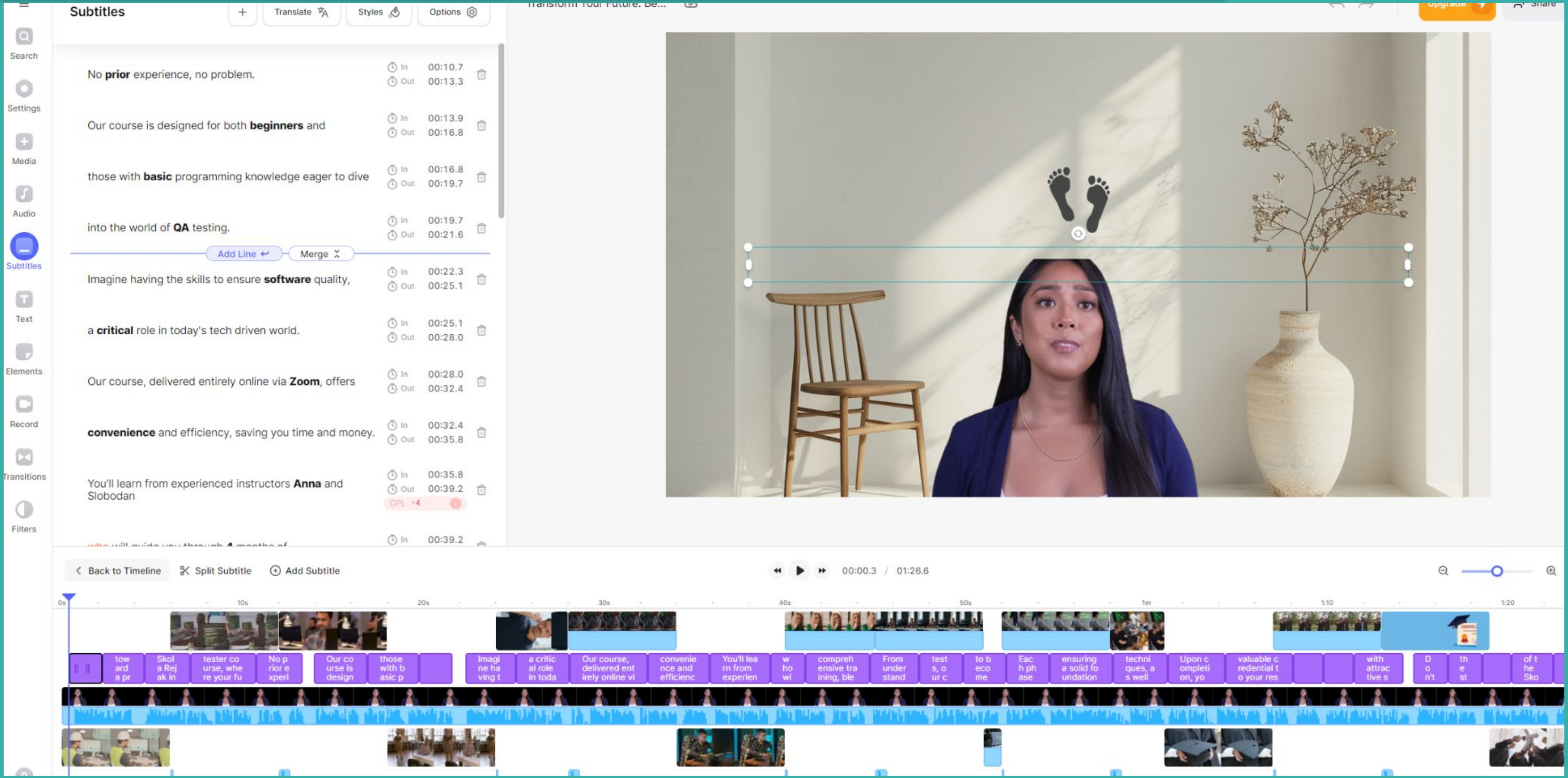
0,500

5,000

Video position: 00.00.34,690

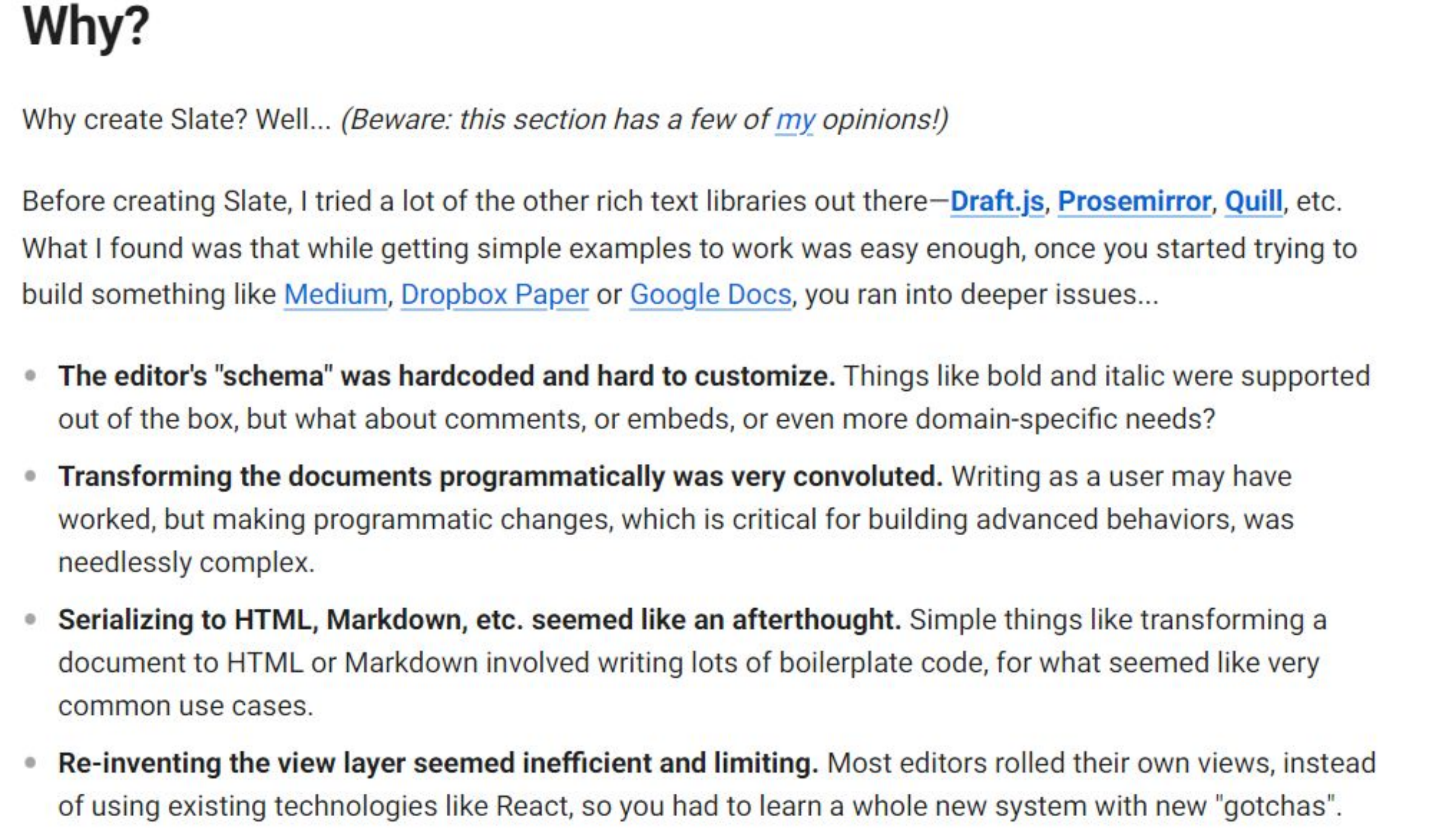
100%

Subtitles Editor



No **prior** experience, no problem.	In 00:10.7	Out 00:13.3
Our course is designed for both **beginners** and	In 00:13.9	Out 00:16.8
those with **basic** programming knowledge eager to dive	In 00:16.8	Out 00:19.7
into the world of **QA** testing.	In 00:19.7	Out 00:21.6
Imagine having the skills to ensure **software** quality,	In 00:22.3	Out 00:25.1
a **critical** role in today's tech driven world.	In 00:25.1	Out 00:28.0
Our course, delivered entirely online via **Zoom**, offers	In 00:28.0	Out 00:32.4
convenience and efficiency, saving you time and money.	In 00:32.4	Out 00:35.8
You'll learn from experienced instructors **Anna** and Slobodan	In 00:35.8	Out 00:39.2 CPL -4
who will guide you through 4 months of	In 00:39.2	

SlateJS



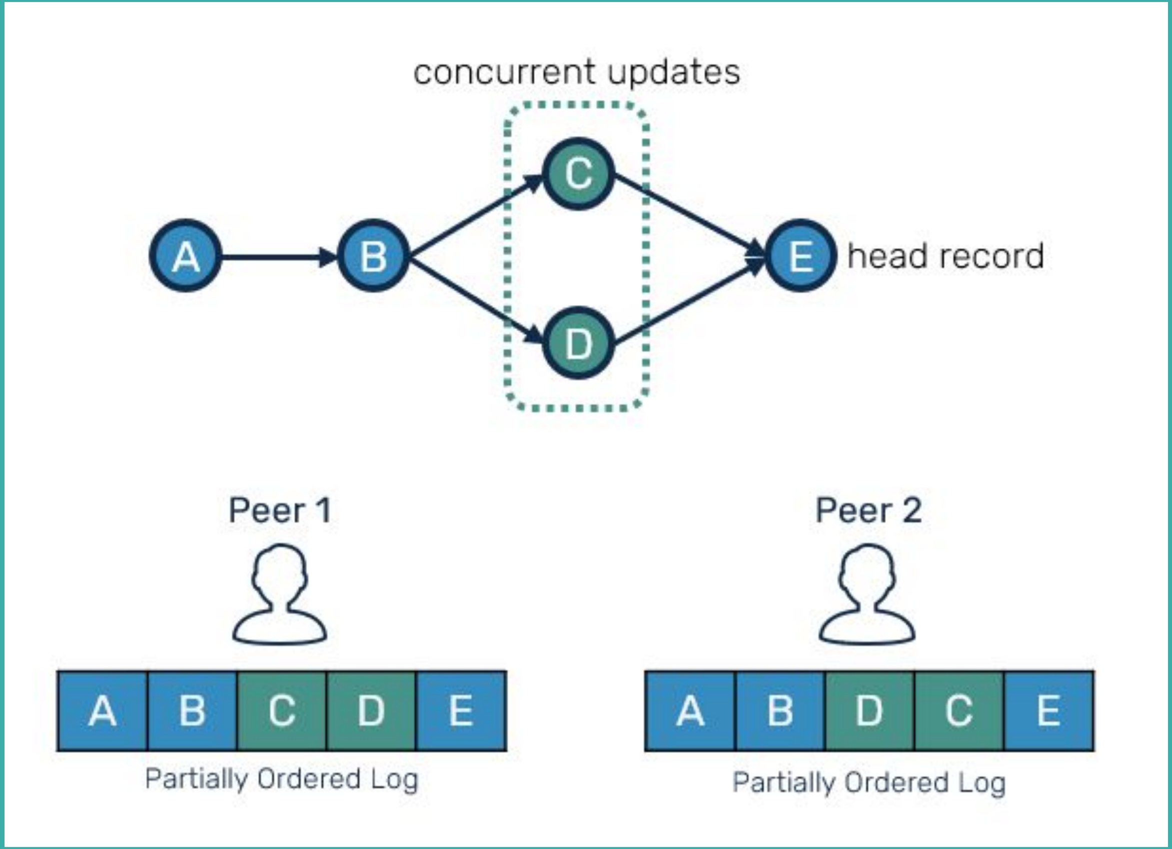
Why?

Why create Slate? Well... *(Beware: this section has a few of my opinions!)*

Before creating Slate, I tried a lot of the other rich text libraries out there—**Draft.js**, **Prosemirror**, **Quill**, etc. What I found was that while getting simple examples to work was easy enough, once you started trying to build something like Medium, Dropbox Paper or Google Docs, you ran into deeper issues...

- **The editor's "schema" was hardcoded and hard to customize.** Things like bold and italic were supported out of the box, but what about comments, or embeds, or even more domain-specific needs?
- **Transforming the documents programmatically was very convoluted.** Writing as a user may have worked, but making programmatic changes, which is critical for building advanced behaviors, was needlessly complex.
- **Serializing to HTML, Markdown, etc. seemed like an afterthought.** Simple things like transforming a document to HTML or Markdown involved writing lots of boilerplate code, for what seemed like very common use cases.
- **Re-inventing the view layer seemed inefficient and limiting.** Most editors rolled their own views, instead of using existing technologies like React, so you had to learn a whole new system with new "gotchas".

Conflict-free Resolution Datatypes



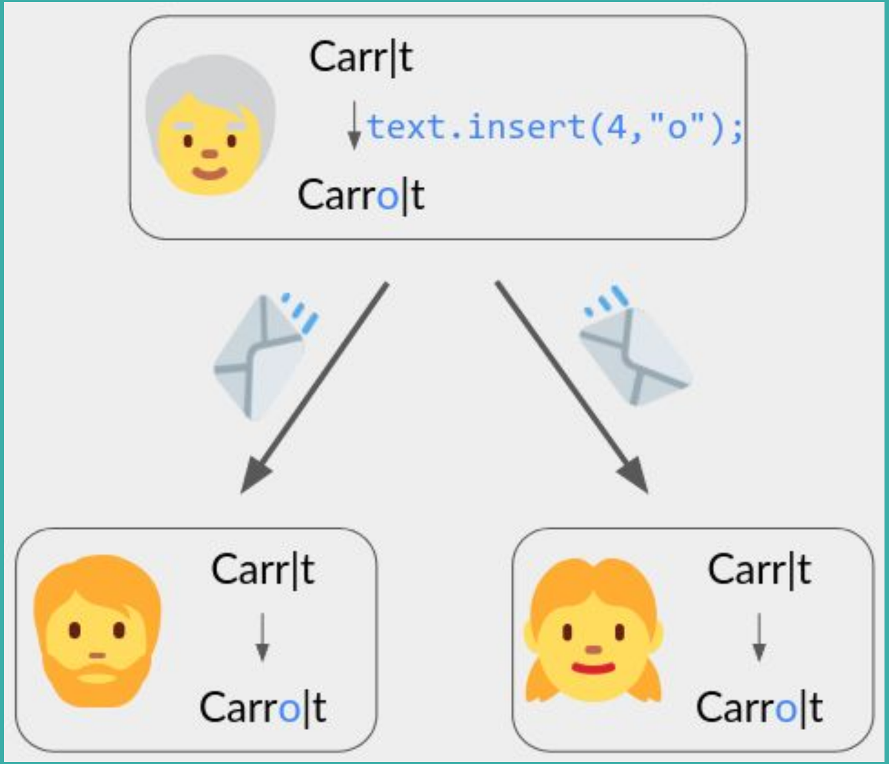
A	B	C	D	E

Partially Ordered Log

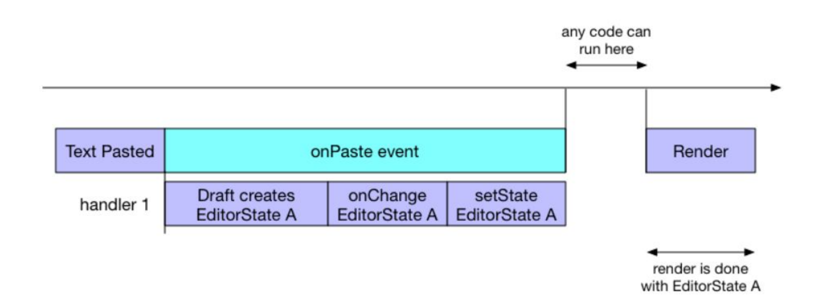
A	B	D	C	E

Partially Ordered Log

Conflict-free Resolution Datatypes



DraftJS (immutable-js)



any code can run here

Text Pasted

onPaste event

Render

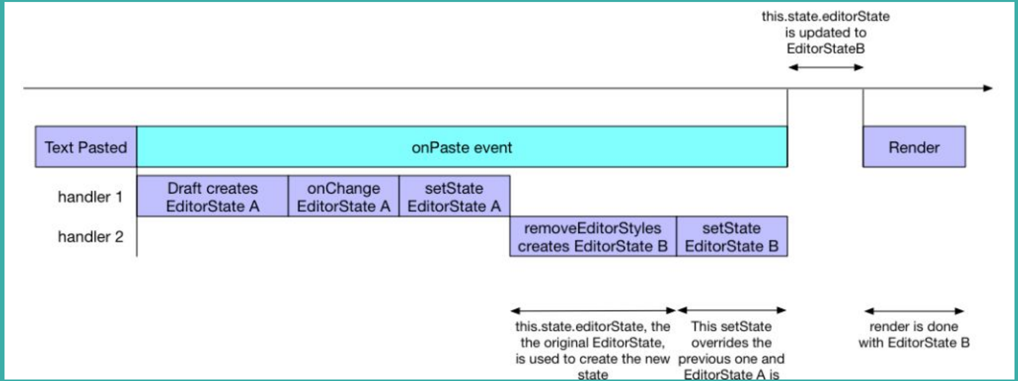
handler 1

Draft creates EditorState A

onChange EditorState A

setState EditorState A

render is done with EditorState A



this.state.editorState is updated to EditorStateB

Text Pasted

onPaste event

Render

handler 1

Draft creates EditorState A

onChange EditorState A

setState EditorState A

handler 2

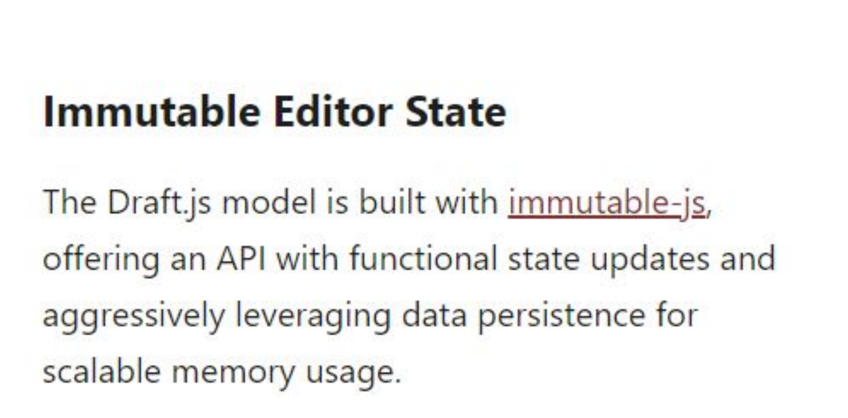
removeEditorStyles creates EditorState B

setState EditorState B

this.state.editorState, the the original EditorState, is used to create the new state

This setState overrides the previous one and EditorState A is

render is done with EditorState B



Immutable Editor State

The Draft.js model is built with immutable-js, offering an API with functional state updates and aggressively leveraging data persistence for scalable memory usage.

Lessons learned

PRODUCT DEV

DEVELOPER'S CLUB

- Roadmap poznat svim DEV ljudima i MENJA SE NA VREME
- **NIKADA BIBLIOTEKE, SVE OD SCRATCH** (za kljucne feature proizvoda)

Hvala!

Postani Član:

developersclub.rs