import React from "https://cdn.skypack.dev/react"

import ReactDOM from "https://cdn.skypack.dev/react-dom"

// Obtengo Provider y connect desde la web

import { Provider, connect } from "https://cdn.skypack.dev/react-redux"

import { createStore, combineReducers, applyMiddleware } from "https://cdn.skypack.dev/redux"

// aquí importamos la librería que define las marcas

import { marked } from "https://cdn.skypack.dev/marked";

// Permitimos los saltos de línea

marked.setOptions({

breaks: true

});

// INSERTS target="\_blank" INTO HREF TAGS (required for Codepen links)

const renderer = new marked.Renderer();

renderer.link = function (href, title, text) {

return `<a target="\_blank" href="${href}">${text}</a>`;

};

const initialText = `

# Welcome to my ReactRedux Markdown Previewer!

## See the next elements:

A link: [FreeCodeCamp](https://www.freecodecamp.org)

A list:

- First level.

- Second level.

A blockcuote: > I think, then i exists.

A bolded text: \*\*This text\*\*

A inline code: \`<div></div>\`

A multi-line code:

\`\`\`

// this is multi-line code:

function anotherExample(firstLine, lastLine) {

if (firstLine == '\`\`\`' && lastLine == '\`\`\`') {

return multiLineCode;

}

}

\`\`\`

And a image:

![Celtics Logo](https://b3h2.scene7.com/is/image/BedBathandBeyond/190604030333\_imageset?$imagePLP$&wid=256&hei=256)

`;

// Redux:

const ADD = 'ADD';

// declaración de una acción que puede ser tratada por messageReducer

const addMessage = (message) => {

return {

type: ADD,

message: message

}

};

// Unifica todas las acciones y estados

const messageReducer = (state = [], action) => {

switch (action.type) {

case ADD:

return [

...state,

action.message

];

default:

return state;

}

};

class Presentational extends React.Component {

constructor(props) {

super(props);

this.state = {

input: initialText

}

// Hacemos el binding a los métodos según el contexto actual

this.handleChange = this.handleChange.bind(this);

}

handleChange(event) {

this.setState({

input: event.target.value

});

}

render() {

return (

<div class="content">

<h1 id="principalTitle">My simple Markdown Previewer</h1>

<textarea id="editor" onChange={this.handleChange}>

{initialText}

</textarea>

<Preview id="preview" input={this.state.input}/>

</div>

);

}

};

const Preview = (props) => {

return (

<div

dangerouslySetInnerHTML={{

\_\_html: marked(props.input, { renderer: renderer })

}}

id="preview">

</div>

);

};

const mapStateToProps = (state) => {

return {messages: state}

};

const mapDispatchToProps = (dispatch) => {

return {

submitNewMessage: (message) => {

dispatch(addMessage(message))

}

}

};

const store = createStore(messageReducer);

const Container = connect(mapStateToProps, mapDispatchToProps)(Presentational);

class AppWrapper extends React.Component {

render() {

return (

<Provider store={store}>

<Container/>

</Provider>

);

}

};

ReactDOM.render(<AppWrapper/>, document.getElementById('root'));