U3SJEDR96 : <@U3KSN5MAL> the code you have on the JS side was fine, only the elm side had to change :slightly_smiling_face:
U3SJEDR96 : I'm guessing you removed the `JSON.parse`?
U3SJEDR96 : I'd go for: `JSON.parse` on the JS side, and passing the resulting object in through flags. On the Elm side, that means you'd take `Json.Encode.Value` as your flags type, and use `Json.Decode.decodeValue` on it to get a `Result String something`
U3KSN5MAL : yeah i got that part working
U3KSN5MAL : sorry should have posted
U3SJEDR96 : yeah, the slashes seem to indicate you're trying to run `decodeValue` on a JS string, basically
U3KSN5MAL : ah
U3KSN5MAL : so pass it in as a json string instead?
U3SJEDR96 : well, I'd try to pass in an actual object
U3SJEDR96 : what does the JS side of things look like, and what does the flags part look like in Elm?
U3KSN5MAL : ```var storedState = localStorage.getItem('LumaPanel');var app = Elm.Main.fullscreen(storedState);```

U3SJEDR96 : right, try making that back into `JSON.parse(localStorage.getItem('LumaPanel')) || {}`
U3KSN5MAL : ```main : Program Json.Value Model Msgmain =
```
    Html.programWithFlags
        { init = init
        , view = view
        , subscriptions = subscriptions
        , update = updateWithStorage
        }
```
```

U3KSN5MAL : ah
U3KSN5MAL : perfect
U3KSN5MAL : working now!
U3SJEDR96 : I mean, you can do it in Elm too, using `decodeString` rather than `decodeValue`, but there's little added value atm
U3KSN5MAL : Yep
U3KSN5MAL : I didn't realise that just changing the types you passed in changed how it interpreted the json
U3KSN5MAL : so i thought i had to just pass in the string
U3KSN5MAL : btw what the new release of my plugin looks like. Totally revamped ui
U3SJEDR96 : yeah, makes sense. But basically, `decodeValue` works on a JS value, `decodeString` works on a string. Any JS value is a valid `Json.Encode.Value`, so passing in a string as a `Json.Encode.Value` means you'd have to `decodeString string flags |&gt; Result.andThen (decodeValue yourDecoder)`
```