

U0NCTKEV8 : (are valid keywords only those keywords that can be produced by the reader?)

U0NCTKEV8 : but generally, turning keys in to keywords all the time is something every project (I have worked on) does, and I hate it

U0NCTKEV8 : but if you do it you are doing the same thing that everyone does

U050MP39D : "valid keywords" are those that follow the rules in the docs imo, but the reader will create some keywords that don't if asked

U0NCTKEV8 : which docs?

U050MP39D : <https://clojure.org/reference/reader#_literals>

U0NCTKEV8 : those a docs on keywords that the reader can read

U0NCTKEV8 : (which is why they are under the reader section)

U0NCTKEV8 : I've seen this discussion play out over and over again in chat, on the mailing list, and in jira issues, and it seems like the core team's general stance is what the reader accepts doesn't define what is valid

U050MP39D : the reader accepts more than those docs, but those docs do give a guarantee of what the reader will accept. so for me they make sense as a definition of valid. not the same thing as "whatever the reader accepts"

U050MP39D : eg``

```
cljs.user=&gt; :&amp;
:&amp;
``
```

```
java.lang.RuntimeException: No reader function for tag foo
clojure.lang.LispReader$ReaderException: java.lang.RuntimeException: No reader function for tag foo
    java.lang.RuntimeException: Unmatched delimiter: )
clojure.lang.LispReader$ReaderException: java.lang.RuntimeException: Unmatched delimiter: )
```

U04V70XH6 : Good question, given that converting arbitrary strings to keywords can open you up to a denial of service attack.

U0NCTKEV8 : I mean, you will likely be fine, and keyword's intern a weakreference, so the they will get cleared

U04V70XH6 : Ya, but... :slightly_smiling_face:

U051SS2EU : ```user=> (defn foo [args] (apply hash-map args))#'user/foo
user=> (binding [*data-readers* {'foo #'user/foo}] (read-string "#foo [1 2 3 4]"))
{1 2, 3 4}```

U3L6TFEJF : <@U0NCTKEV8> ahh, so you need this? `(binding [*data-readers* {'foo #'user/foo}] (read-string "#foo [1 2 3 4]"))`

U3L6TFEJF : haha, timing :smile:

U0NCTKEV8 : yes

U3L6TFEJF : while on the topic of tagged literals: is there anything special you need to do to get it to work with
`tools.namespace`?