

```

U0W0JDY4C : so even if I invoke `(protocols/foo this)`, it still doesnt matter if I refer to the record's field or the `(:foo this)` because at that point the context has already changed
U0NCTKEV8 : what?
U0NCTKEV8 : the first argument is always going to be the thing you invoked the protocol function
U0NCTKEV8 : on
U0W0JDY4C : ```(defrecord MyRecord [field]
  IMyProto
  (foo [this a] (do-something field)) &lt;- here shouldnt matter if I use field or (:field this)?
  (bar [this a] (protocols/foo this a)
...

```

```

U0W0JDY4C : I just want to make sure I dont get this very subtle bug by implementing a protocol that's using a field
versus a lookup on this, when self-referencing(?) a protocol
U050MP39D : `field` and `(:field this)` will always be the same thing
U0W0JDY4C : i've been stabbed in the foot by javascript so many times I'm very paranoid
U0W0JDY4C : okay, good to know the context stays the same no matter how it's invoked
U0W0JDY4C : thanks <@U0NCTKEV8> <@U050MP39D> <@U051SS2EU> :thumbsup:
U0W0JDY4C : ```let my_obj = {
  lolGoodLuck(args) {
    let that = this
    function() {
      that.call(this, args)
    }
  }
}
my_obj.lolGoodLuck.apply(youWillNeverGuess)
...

```

```

U17DY48BW : Got a tough problem if anyone has an answer. Basically I have a large data structure (vector of
hashmaps) about 50,000 hashmaps. And I need to pass the data to a Java function, but the hashmap has keyword
keywords and those keywords need to be transferred to strings for the java function. Is there a fast way to convert
keyword keywords to strings? right now the conversion takes about 2 seconds
U07S8JGF7 : Use strings from the start instead of keywords?
U07S8JGF7 : You're talking linear time + string manipulation. Probably not a fast operation no matter how you slice it.
U051SS2EU : name shouldn't be manipulating strings - it should just be getting a field from the keyword
U07S8JGF7 : Turning string data into keywords is a bad habit that clojure developers have. Benefit is really only
concision, downsides are slow and buggy translation.
U051SS2EU : (I mean, if you are converting by manipulating strings, use name instead)
U07S8JGF7 : Yeah right. Fair enough.
U1ALMRBLL : <@U17DY48BW> I just generated 50,000 keywords, at 50 random characters each (using c.s.gen), and
both `str` and `name` take about &lt;10ms to convert the whole vector from keywords to strings
...

```

```

(defn trunc [s n]
  (subs s 0 (min (count s) n)))

(def kws
  (doall
    (gen/sample
      (gen/fmap
        #(keyword (trunc (name %) 50))
        (gen/keyword))
      50000))

(time (do (doall (map name kws)) nil))
"Elapsed time: 5.684874 msecs"
=> nil
(time (do (doall (map str kws)) nil))
"Elapsed time: 5.781799 msecs"
=> nil
...

```

U1ALMRBLL : (where `gen` is `clojure.spec.gen.alpha`)

U17DY48BW : <@U1ALMRBLL> ya your test doesn't compare as I am talking about a series of hashmaps with multiple keys in each one

U17DY48BW : but <@U07S8JGF7> your suggestion is what I landed on, so I'll just change my json decoding to get string keys

U1ALMRBLL : oh, so there are many such keywords in each hashmap?

U17DY48BW : <@U1ALMRBLL> correct

U1ALMRBLL : how many total?

U051SS2EU : I bet the map construction when converting is the more expensive part, not the keyword/string part

U1ALMRBLL : ^^ yep

U051SS2EU : but avoiding converting avoids both