

Sharing specs and ns

Define whisky spec with namespaced key

```
(ns whisky-spec.core  
  (:require [clojure.spec.alpha :as s]))  
  
(s/def :liquor/whisky (s/keys :req [::name ::age]))
```

Use whisky spec

```
(ns liquor-shop.core
  (:require [clojure.spec.alpha :as s]
            [whisky-spec.core])) ;; ??
```

```
(defn whisky? [whisky-map]
  (s/valid? :liquor/whisky whisky-map))
```

Problem: no idea where :liquor/wisky comes from

s/registry is global

```
(:liquor/wisky (s/registry))  
nil
```

```
(require 'wisky-spec.core)  
(:liquor/wisky (s/registry))  
#object[clojure.spec.alpha$map_spec_impl  
$reify__1931 0x765d259b "clojure.spec.alpha  
$map_spec_impl$reify__1931@765d259b"]
```

Problem: name conflicts

I prefer :: with alias

```
(ns whisky-spec.core  
  (:require [clojure.spec.alpha :as s]))
```

```
(s/def ::whisky (s/keys :req [::name ::age]))
```

```
(ns liquor-shop.core  
  (:require [clojure.spec.alpha :as s]  
            [whisky-spec.core :as ws]))
```

```
(defn whisky? [whisky-map]  
  ;; (s/valid? :whisky-spec.core/whisky whisky-map)  
  (s/valid? ::ws/whisky whisky-map))
```

Problem: over the wire


```
(ns liquor-app.core
  (:require [clojure.spec.alpha :as s]
            [whisky-spec.spec :as ws]))
```

```
(defn sell-whisky [name]
  (let [whisky (get-whisky name)]
    (if (s/valid? :ws/whisky whisky)
        (sell-to-customer whisky)
        (throw-away! whisky)))))
```

`{:whisky-spec.spec/name "suntry"`
`:whisky-spec.spec/age 10}`



```
(ns alcohol-service.core)
```

```
(defn send-whisky [name age]
  (let [whisky {:ws/name "suntry"
                :ws/age 10}]
    (deliver-whisky whisky)))
```