```clojure
:worker-threads :port :host :ssl-context :io-threads :client-auth :ajp-port :direct-buffers? :trust-password
```

```clojure
#{:path :dispatch? :trust-managers :key-managers :keystore :buffer-size :auto-start :buffers-per-
```

`:virtual-host :key-password :truststore :configuration :contexts :http2? :servlet-name :filter-map :ssl-`

```clojure
(immutant/run (wrap-component handler m) (select-keys m valid-options))
```

```
(timbre/debug "Launching Immutant web server.")
```

```clojure
(defmethod ig/init-key ::server [_ {:keys [handler] :as m}]
```

```clojure
(timbre/debug "Stopping Immutant web server.")
```

```clojure
(defmethod ig/halt-key! ::server [_ server]
```

```
(def valid-options
```

region :static-dir

```
    (immutant/stop server))
```

port})

```clojure
(defn wrap-component [handler component]
```

```clojure
(fn [request] (handler (into component request))))
```

```clojure
:worker-threads :port :host :ssl-context :io-threads :client-auth :ajp-port :direct-buffers? :trust-password
```

```clojure
#{:path :dispatch? :trust-managers :key-managers :keystore :buffer-size :auto-start :buffers-per-
```

`:virtual-host :key-password :truststore :configuration :contexts :http2? :servlet-name :filter-map :ssl-`

```clojure
(immutant/run (wrap-component handler m) (select-keys m valid-options))
```

```clojure
(timbre/debug "Launching Immutant web server.")
```

```clojure
(defmethod ig/init-key ::server [_ {:keys [handler] :as m}]
```

```clojure
(timbre/debug "Stopping Immutant web server.")
```

```clojure
(defmethod ig/halt-key! ::server [_ server]
```

```clojure
(defmethod ig/halt-key! ::server [_ server]
```

```
(def valid-options
```

region :static-dir

```
(immutant/stop server))
```

port})