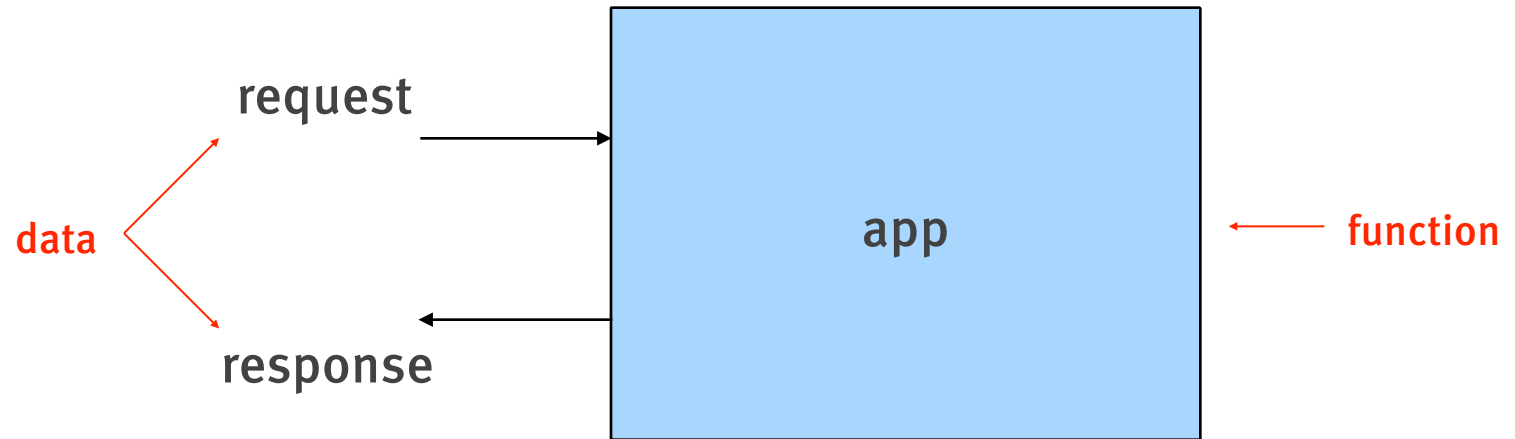


Web Development?



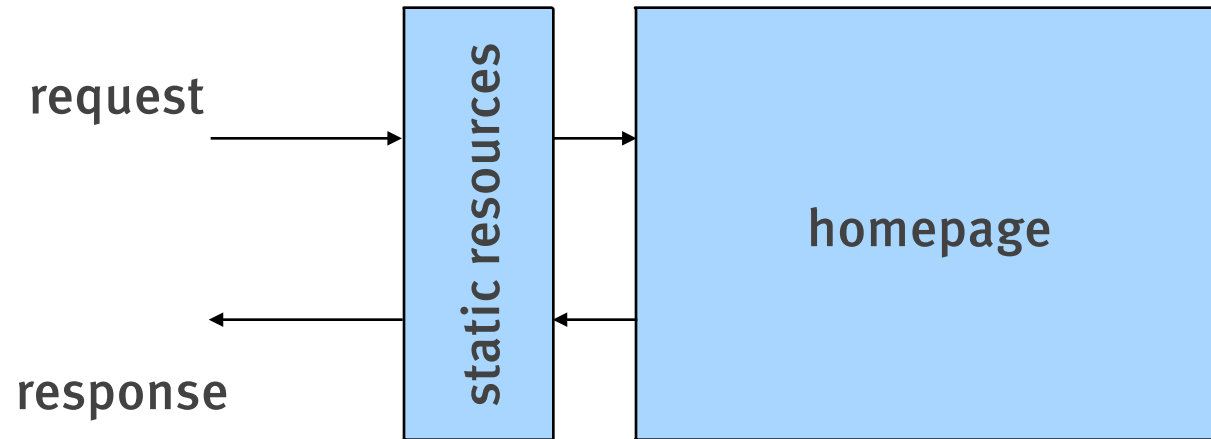
Ring

```
(defn hello-world-app [req]
  {:status 200
   :headers {"Content-Type" "text/plain"}
   :body "Hello, World!"})
```

```
(hello-world-app {:uri "/foo"
                  :request-method :get})
> {...}
```

```
(run-jetty hello-world-app {:port 8080})
```

```
(defn my-first-homepage [req]
  {:status 200
   :headers {"Content-Type" "text/html"}
   :body (str "<html><head>"
              "<link href=\"/pretty.css\" ...>"
              "</head><body>"
              "<h1>Welcome to my Homepage</h1>"
              (java.util.Date.)
              "</body></html>"))})
```



```
(defn decorate [webapp]  
  (fn [req]  
    ...before webapp...  
    (webapp req)  
    ...after webapp...))
```

```
(defn decorate [webapp]
  (fn [req]
    (if (static-resource? req)
      (return-resource req)
      (webapp req)))))
```



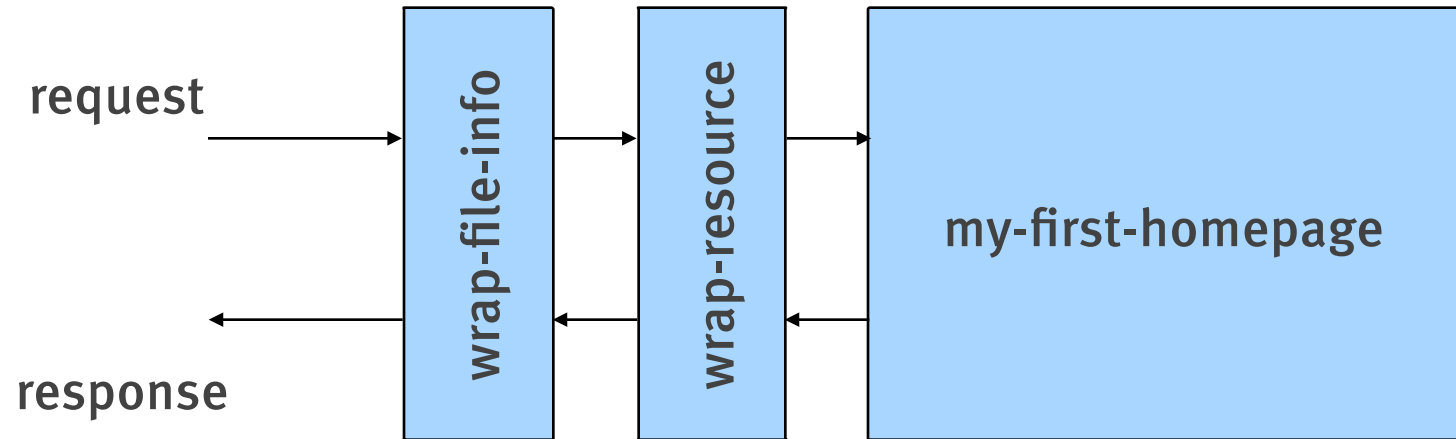
```
(defn wrap-resource [handler root-path]
  (fn [request]
    (if-not (= :get (:request-method request))
      (handler request)
      (let [path (extract-path request)]
        (or (resource-response path {:root root-path})
            (handler request)))))))
```

```
(defn my-first-homepage [req] ...)
```

```
(def webapp  
  (wrap-resource my-first-homepage "public"))
```

```
(run-jetty webapp {:port 8080})
```

```
(webapp {:uri "/pretty.css"  
        :request-method :get  
        :headers {}})  
> {:status 200  
   :headers {}  
   :body #<File ...resources/public/pretty.css>}
```



```
(defn homepage [req] ...)
```

```
(def webapp  
  (-> homepage  
        (wrap-resource "public")  
        wrap-file-info))
```

```
(wrap-file-info  
  (wrap-resource  
    homepage  
    "public"))
```

```
(run-jetty webapp {:port 8080})
```

```
(webapp {:uri "/pretty.css"
          :request-method :get
          :headers {}})
> {:status 200
   :headers {"Content-Length" "16"
             "Last-Modified" "Thu, 14 Jun ..."
             "Content-Type" "text/css"}
   :body #<File ...resources/public/pretty.css>}
```

wrap-resource

wrap-file

wrap-params

wrap-session

wrap-flash

wrap-etag

wrap-basic-authentication