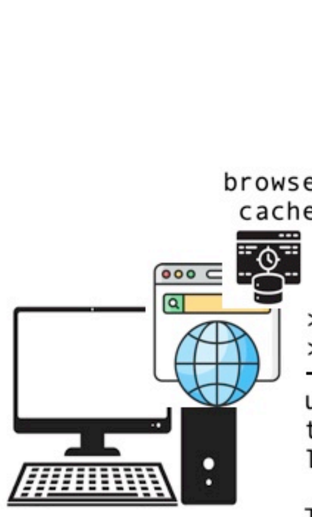

Cache-Control: max-age=1000*60,s-max-age=1000*60*24

```
swiggy (webproject)
|-src
  |-main
    |-java
    |-resources
      |-static
        |-css/        ——protected resources
        |-js/
        |-images/
        |-fonts/
    |-webapp
      |-WEB-INF
        |-html
          |-*.html
|-pom.xml
|-targe
```

[war]

Client-Side Caching: cache-control header

>(1. request): /index.htm
>(2. request): /about-us.htm

upon receiving the request, the webserver serves the corresponding resource that has been requested like index.html or about-us.htm

The Web browser while parsing and rendering the HTML content, it identifies the references to these static resources like css, js, images, fonts etc, and spans requests to the web server asking to download these resources inorder to render the web page.

each web page in our application will have references to the static resources

```
index.html
<html>
  <head>
    <link rel=“stylesheet” href=“/css/bootstrap.min.css”>
    <link rel=“stylesheet” href=“site.css”>
  </head>
  <body>
    <img src=“images/logo.jpg”/>
    <script type=“text/javascript” src=“js/bootstrap.min.js”/>
  </body>
</html>
```

While rendering every webpage of our application by the web browser, it will repeated sends a bunch of requests in downloading the static resources that are referenced in our web pages, since these static resources are seems to be constant and same across for any number of requests, repeatedly serving these resources to the client leads to several problems:

1. the network traffic/congession between the browser/client and the webserver would be high that results in latency in serving the requests
2. bandwidth consumption would be unnecessarily increases in serving repeatedly the same resources to the client
3. Most of the I/O Capacity of the WebServer would be occupied in serving the static resources thus significantly slowing down the performance of the server.

How to solve this problem in serving the static resources of our application to the clients?
There are #2 solutions in addressing this problem
1. CDN Proxies (AWS CloudPlatform: CloudFront Service)
2. Client-Side Caching

The application has to instruct the CDN Server or Client (browser) asking them to cache the static resources for central amount of time by sending Cache-Control header aspart of the HttpResponse.

How does the application should instruct the client/intermediatory in caching the static resources?
1. dont let the client to access the static resources directly, place them inside the protected directory of the application

```
src
|-main
  |-resources
    |-static
      |-css
      |-js
      |-images
…
```

2. inorder to serve these resources we need to have an common ResourceController that would receive the request on a wildcart pattern like /static/** or /public/** and lookup for these resources within our application and dispatch them aspart of the response by including Cache-Control header aswell.

Having a resource controller in dispatching the static resources seems to be an common requirement and endup in writing the boiler-plate logic across the applications, so spring framework has provided ResourceHandlers to takecare of dispatching these static resources as below.

```
@Configuration
@EnableWebMvc
class WebMvcConfig implements WebMvcConfigurer {

  public void addResourceHandlers(ResourceHandlerRegistry registry) {
    registry.addResourceHandler(“/static/**”).addResourceLocations(“classpath:/static/“).setCachePeriod(1000*60*60);
  }
}
```

http://localhost:8080/swiggy/static/css/site.css -> Now the request will be received by the ResourceHandler and lookup for the resource under classpath:static directory and dispatches the resource along with Cache-Control header asking the client to cache.