

گزارش تمرین اول مهندسی اینترنت

تینا صداقت ۹۳۳۱۰۴۴

۱. اطلاعات نرم افزار مورد استفاده شده برای وب سرور

از تابع `getHeaderField()` استفاده می کنیم. به این صورت که با دادن نام `Server` به عنوان آرگومان، اگر مقداری برای آن `Set` شده باشد را `return` می کند.

```
38. public StringBuilder serverInformation() {  
39.     answer1.append("Server: " + connection1.getHeaderField("Server") + "\n");  
40.     return answer1;  
41. }
```

۲. لیست متدهای HTTP که سرور بر روی URL مد نظر می پذیرد

می دانیم تابع `OPTIONS` لیست تابع های قابل پذیرش روی `URL` مدنظر را می دهد. پس به کمک تابع `setRequestMethod()` تابع `OPTIONS` را صدا می زنیم و به کمک `getHeaderField("Allow")` لیست تابع های قابل پذیرش را نمایش می دهیم.

```
43. public StringBuilder SupportedServerMethods() throws IOException{  
44.     connection2 = (HttpURLConnection) url.openConnection();  
45.     connection2.setRequestMethod("OPTIONS");  
46.     String SupportedMethods = connection2.getHeaderField("Allow");  
47.     answer2.append("Supported Methods: " + SupportedMethods + "\n");  
48.     return answer2;  
49. }
```

۳. لیست cookie هایی که توسط سرور `set` می شود. به همراه اطلاعات هر `cookie` شامل اسم

و مقدار و مسیر و `domain` و زمان `expire` شدن و `HTTP only` و امنیت

ابتدا به کمک تابع `getHeaderField()` قسمت مربوط به `Set-Cookie` را پیدا می کنیم. در رشته ای که درمورد `cookie` ها به دست می آید، هر قسمت به کمک `;` جدا شده است که برای واضح تر خوانده شدن اطلاعات قسمت هایی که با `;` جدا شده است را در خط جدید می نویسیم.

```

51. public StringBuilder cookieInformation() throws IOException{
52.     String cookies = connection1.getHeaderField("Set-Cookie");
53.     if(cookies!=null && cookies.contains(";")) {
54.         cookies=cookies.replaceAll(";", "\n");
55.         answer3.append("cookies" + cookies + "\n" );
56.         if(cookies.contains("secure")) {
57.             //answer3.append(" is secure= true");
58.         }
59.         else
60.             answer3.append("is secure=false");
61.     }
62.     else
63.         answer3.append("cookie is null");
64.
65.     return answer3;
66. }

```

۴. اطلاعات مربوط به cache شامل expire و last-modified و cache control

باز از تابع `getHeaderField()` برای به دست آوردن اطلاعات `cache` شامل `Expires` و `Last-Modified` و `Cache-Control` استفاده می کنیم.

```

67. public StringBuilder cacheInformation() throws IOException {
68.     headerFields = connection1.getHeaderFields();
69.     answer4.append("Expires: " + headerFields.get("Expires") + "\n" + "Last-
        Modified: " + headerFields.get("Last-Modified") + "\n" + "Cache-
        Control: " + headerFields.get("Cache-Control") + "\n");
70.     return answer4;
71. }

```

۵. اگر برای دسترسی به آن URL نیاز به Authentication است، نوع Authentication و اطلاعات مربوطه

برای به دست آوردن اطلاعات مربوط به Authentication، در `headerFields` دنبال عبارت `Proxy-Authenticate` می گردیم.

```

1. public StringBuilder authenticationInformation() throws IOException {
2.     headerFields = connection1.getHeaderFields();
3.     answer5.append("Authentication: " + headerFields.get("Proxy-
        Authenticate") + "\n");
4.     return answer5;
5. }

```

۶. نوع خطا و توضیحات آن در صورتی که سرور در پاسخ به URL داده شده پیغام خطا تولید کند

از تابع `getResponseCode()` برای نوع خطا و از تابع `getResponseMessage()` برای توضیح خطا استفاده می کنیم.

```
1. public StringBuilder getStatusCode() {  
2.     try {  
3.         answer6.append(+ connection1.getResponseCode() + " " +connection1.getResponse  
         Message());  
4.         catch (IOException e) {  
5.             e.printStackTrace();  
6.         }  
7.         return answer6;  
8.     }
```

۷. پشتیبانی از Persistent Connections

برای این کار از تابع `setRequestProperty()` استفاده می کنیم و کلید را `connection` و مقدار را `keep-alive` می گذاریم.

```
20. connection1.setRequestProperty("Connection", "keep-alive");
```

پیاده سازی قسمت واسط کاربری:

به کمک کلاس `Application` که از `JavaFX` ، `extend` می شود قسمت `user interface` را پیاده کردیم. یک `GridPane` داریم که جواب سوال های مطرح شده را در هر سطر این جدول قرار می دهیم. از `textview` برای قسمت گرفتن URL از کاربر و از `button` به عنوان دکمه استفاده شده است. با کلیک کردن کاربر روی دکمه ی `analyze` تابع ها صدا زده می شوند و در یک `ListView` قرار می گیرند و نمایش داده می شوند.