Gate-Keeper Senior Test

1. Core Tech stack

- Spring Configuration Processor
- Spring Web
- Spring Web Services
- Apache Freemarker
- Spring Security
- Spring Data MongoDB

2. Start

- Checkout the code from:

  https://github.com/HariRangarajan-Accenture/accenture-access-gateway

- Install "mvn clean install"
- Run backend "mvn spring-boot:run"
- Confirm application.properties for user login details

3. Basic Description:

- Users need to login (authenticated) and before they are allowed to **access protected resources**.
- Modify the **BE** to support the **missing functionality**.

4. Back-end

- **Session management (missing functionality)**
  - **Each request should get session GRANT cookie**, only once session is authenticated can the grant be linked to user.
- **Your BE should at least have 3 categories of static resources which can be accessed depending on access level (Available)**
  - **MISSING * anonymous (0)** required resource or page *
  - **low access (1)** required resource or page
  - **high access (2)** required resource or page
- **Bonus Points:**
  - Identify and document any security related issues, provide possible solutions in the documentation.

Tasks

- improve the code as much as possible, logging etc

- add missing functionality
- find and fix bugs

How to:

Authenticate user

GET "http://localhost:8080/access"

- Authorization header is used for user login
- Input example Authorization: Basic {requested_level}#{username}:{password} encoded base64

Or GET request via browser can be made as shown by example below:

Input for username "{access-level}#{username}"

Any request to a resource (static/Level{N}/low_access.txt) requires that XSRF header is set, this header must match the cookie value of CSRF returned on authentication

5. Front-End (Bonus section)

Build **FE** with minimal effort(for example using FE frameworks/libraries) which will do an authentication request based on what **error** is **returned** from the **BE**.

On loading the "/home", the front end should be presented, showing all available resources, once user clicks on a link the fronted should react to the response and present the appropriate authentication screen. Once the user is authenticated the original requested resource should be displayed.

Access violation example

```
{"requiredAccess":"Level1","message":"invalid access level"}
```

Loading front-end

- "resources/templates/angular-app.ftl" as your angular app loading point (may be any front-end framework)

Your FE should support 2 types of authentication which can result in different levels of access

- (low access) numeric code authentication
- (high access) alpha numeric

Available resources

- GET "http://localhost:8080/resources"

```
{
   "resources":[
      "/Level1/low/access.txt",
      "/Level1/low_access.txt",
      "/Level2/high_access.txt",
      "/Level2/what/am/I/access.txt",
      "/css/main.css",
      "/js/main.js"
   ]
}
```

Flow examples of FE&BE behaviour combined

- Static resources/pages which are behind access restriction are inaccessible directly. (Provided from BE)
- When user with no access tries to access **low access** page then authentication is required. (Provided from BE)
- When user has **low access** and needs to access **high access** page then the user needs to be authenticated with **high access** and redirected to the static resource/page. (FE needs to do routing for user journey)
- When request is made to **no access** static resource then the resource is returned to user. (Provided by BE)

6. Known startup issues:

Running mvn clean install or mvn spring-boot:run you may run in to some errors like below

```
Exception in monitor thread while connecting to server
localhost:27017com.mongodb.MongoSocketOpenException: Exception opening socket
        at
com.mongodb.internal.connection.SocketStream.open(SocketStream.java:70)
~[mongodb-driver-core-3.11.2.jar:na]
        at
com.mongodb.internal.connection.InternalStreamConnection.open(InternalStreamC
onnection.java:128) ~[mongodb-driver-core-3.11.2.jar:na]
        at
com.mongodb.internal.connection.DefaultServerMonitor$ServerMonitorRunnable.ru
n(DefaultServerMonitor.java:117) ~[mongodb-driver-core-3.11.2.jar:na]
        at java.lang.Thread.run(Thread.java:748) [na:1.8.0_221]
Caused by: java.net.ConnectException: Connection refused (Connection refused)
        at java.net.PlainSocketImpl.socketConnect(Native Method)
~[na:1.8.0_221]
        at
java.net.AbstractPlainSocketImpl.doConnect(AbstractPlainSocketImpl.java:350)
~[na:1.8.0_221]
        at
java.net.AbstractPlainSocketImpl.connectToAddress(AbstractPlainSocketImpl.jav
a:206) ~[na:1.8.0_221]
```

```
        at
java.net.AbstractPlainSocketImpl.connect(AbstractPlainSocketImpl.java:188)
~[na:1.8.0_221]
        at java.net.SocksSocketImpl.connect(SocksSocketImpl.java:392)
~[na:1.8.0_221]
        at java.net.Socket.connect(Socket.java:589) ~[na:1.8.0_221]
        at
com.mongodb.internal.connection.SocketStreamHelper.initialize(SocketStreamHel
per.java:64) ~[mongodb-driver-core-3.11.2.jar:na]
        at
com.mongodb.internal.connection.SocketStream.initializeSocket(SocketStream.ja
va:79) ~[mongodb-driver-core-3.11.2.jar:na]
        at
com.mongodb.internal.connection.SocketStream.open(SocketStream.java:65)
~[mongodb-driver-core-3.11.2.jar:na]
        ... 3 common frames omitted
```

Publish your web app on your github repo and send us a link to review.

We expect you to present your changes and flow during the technical discussion which will be scheduled next.

Accenture does not hold any rights on this codebase.