Joseph Boylan  
Service API Design and Development   
Assignment 11

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Text

Description automatically generated

\*A side note, I was unable to get the PUT and DELETE requests to work with authentication\*

**Step 2:**

The repository classes are interface classes for each of the model classes that contain an analogy of the SQL language which allows the query methods to be made. Each one will be able to find, add, post, and delete information about the objects from the database.

The service classes are interface classes that provide the structure of the logic for login which takes parameters that are part of the process of generating tokens. Another is taking in the token as an input parameter. Lastly, the ability to find the student by id.

The Service Implementation classes are used to implement the Service classes. It’s a RESTful service for each of the objects that contains three methods, login, findByToken, and findByID. The login method generates and returns the token. findByToken and findByID take in the token or ID to access the database information through the request methods.

The AuthenticationProvider, AuthenticationFilter, and SecurityConfiguration classes are the three classes directly aligned with allowing authentication through Spring Security.

AuthenticationProvider is a spring boot class that contains a method that takes in parameters username and UsernamePasswordAuthenticationToken class and is used to return the user’s credentials.

AuthenticationFilter is a spring boot class and is a filter class that intercepts incoming traffic and navigates traffic to the appropriate processor in case additional authentication is needed. It overrides two methods, attemptAuthentication, and successfulAuthentication. AttemptAuthentication extracts the token as part of the user's input and uses it as a parameter to use the AuthenticationManager during the request. SuccessfulAuthentication sends results to a SecurityContextHolder and makes a call to a chain filter.

SecurityConfiguration class is responsible for the configuration of the entire process of authentication in Spring Security. It includes a matcher object that makes any URL that contains /api/\* to be subject to authentication. Additional configuration methods are included to add additional configurations to Spring Security.