Jwt Authentication Controller

Path: com.opus.backend.controller.JwtAuthenticationController.class

This class handles various API endpoints related to authentication, user registration, password management, account verification, and subscription management. Here's a breakdown of each function in the code:

1. **createAuthenticationToken(JwtRequest authenticationRequest)**: This function takes a JWT request containing a username and password. It authenticates the user and, if successful, generates a JWT token and returns it as a response. It also checks if the user's account is verified and of the correct account type.
2. **saveUser(UserDTO user)**: This function takes a user DTO (Data Transfer Object) and registers a new user. It checks if the user or email already exists, sends a verification email to free users, and sends a registration confirmation email to partner users and the confirmation will be from Opus side.
3. **ForgetPassword(String username)**: This function takes a username (or email) and sends a password reset email to the user with a reset link.
4. **ResetPassword(String token, String password)**: This function takes a JWT token and a new password, extracts the username from the token, and updates the user's password in the database.
5. **authenticate(String username, String password)**: This function is a utility method that authenticates a user using the provided username and password. It throws exceptions if the user is disabled or the credentials are invalid.
6. **hello(String token)**: This function takes a JWT token and verifies the user associated with the token. It updates the user's verification status in the database and sends a confirmation email if the user is a partner.
7. **Subscribe(String username, String subscription\_expiration\_date)**: This function takes a username (or email) and a subscription expiration date, updates the user's subscription details, and sends a subscription confirmation email.
8. **UserDetails(String username)**: This function takes a username (or email) and returns the user's details, including the username, email, company name, subscription start date, and subscription expiration date.

These functions are exposed through API endpoints using the **@RequestMapping** annotation with appropriate HTTP methods (GET or POST) and path values (e.g., **/authenticate**, **/register**, etc.).