Okta OIDC Client Creation and Configuration for Spring Boot

1. Creating an OIDC Client in Okta

1. Log in to your Okta Developer Console.
2. Navigate to **Applications** > **Applications**.
3. Click on **Create App Integration**.
4. Select **OIDC - OpenID Connect** as the Sign-in method.
5. Choose **Web Application** and click **Next**.
6. Fill in the details:
    - App Name: `SpringBootApp`
    - Sign-in redirect URIs: `http://localhost:8080/login/oauth2/code/okta`
    - Sign-out redirect URIs: `http://localhost:8080`
7. Assign users or groups as needed.
8. Click **Save** and copy the **Client ID** and **Client Secret**.

2. Configuring Default Scopes and Policies

1. Navigate to **Security** > **API** > **Authorization Servers**.

2. Select the **default** authorization server.

3. Go to the **Scopes** tab and add the following scopes if not present:

    - `openid`
    - `profile`
    - `email`
4. Go to the **Access Policies** tab:

    - Create a new policy if needed.
    - Add a rule to allow access to your OIDC client.

3. Spring Boot Integration

Dependencies

Add the following dependencies to `pom.xml`:

```
<dependencies>
  <dependency>
```

```
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-oauth2-client</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-security</artifactId>
    </dependency>
    <dependency>
        <groupId>org.springframework.boot</groupId>
        <artifactId>spring-boot-starter-web</artifactId>
    </dependency>
</dependencies>
```

Spring Security Filter Configuration

Create a security configuration class:

```
import org.springframework.context.annotation.Bean;
import org.springframework.security.config.annotation.web.builders.HttpSecurity;
import org.springframework.security.web.SecurityFilterChain;

@Configuration
public class SecurityConfig {
    @Bean
    public SecurityFilterChain securityFilterChain(HttpSecurity http) throws Exception {
        http
            .authorizeHttpRequests(auth -> auth
                .requestMatchers("/").permitAll()
                .anyRequest().authenticated()
            )
            .oauth2Login();
        return http.build();
    }
}
```

Application Properties Configuration

Add the following to `application.properties`:

```
spring.security.oauth2.client.registration.okta.client-id=your-client-id
spring.security.oauth2.client.registration.okta.client-secret=your-client-secret
spring.security.oauth2.client.registration.okta.scope=openid, profile, email
```

spring.security.oauth2.client.provider.okta.issuer-uri=https://your-okta-domain/oauth2/default

4. Running and Testing

1. Start the Spring Boot application.
2. Access `http://localhost:8080`, and you should be redirected to Okta for authentication.
3. Upon successful login, Okta redirects back to the application with user authentication.

This setup ensures secure authentication and authorization for your Spring Boot application using Okta OIDC.