

## SSO Integration with Azure

March 06, 2024

### Tech Stack:

Microsoft Azure, ReactJS, Java

## Product Overview

Single Sign-On (SSO) integration with Azure enables seamless authentication and access control across various applications within an organization. This Proof of Concept (PoC) aims to demonstrate the implementation of SSO with Azure using Java for backend services and React JS for frontend applications.

### Why are we doing this?

Integrating Azure using Single Sign-On (SSO) offers a myriad of advantages that significantly enhance both user experience and organizational security. By consolidating authentication processes, SSO eliminates the need for users to juggle multiple sets of credentials across various applications. This streamlined approach not only simplifies user access but also reduces the likelihood of password fatigue and enhances overall user satisfaction. Furthermore, from an administrative standpoint, SSO centralizes access management, enabling administrators to efficiently provision and deprovision user access to multiple resources from a single location. This not only saves time but also



enhances security by ensuring consistent enforcement of access policies and reducing the risk of unauthorized access. Moreover, SSO enhances security by providing a robust authentication framework, which supports multi factor authentication and other advanced security measures. This ensures that access to sensitive resources is protected against unauthorized access attempts, thereby bolstering the organization's security posture. Overall, integrating Azure with SSO not only simplifies user access but also strengthens security measures, thereby facilitating a more efficient and secure organizational environment.

## What Are We Gonna Do?

The Single Sign-On (SSO) Integration with Azure using Java and React JS aims to seamlessly authenticate users across multiple applications within an organization using Azure Active Directory (Azure AD). Our solution will provide a robust and secure authentication mechanism while enhancing user experience and access management.

### Key Features:

- Seamless Authentication:
  - Implement SSO with Azure AD to enable users to access multiple applications with a single set of credentials.
- Secure Backend Services:
  - Develop a Java backend service using Spring Boot to handle authentication with Azure AD and secure backend APIs.
- User-friendly Frontend:
  - Create a React JS frontend application integrated with Azure AD SSO to provide a smooth authentication experience for users.
- Centralized Access Management:
  - Utilize Azure AD to centrally manage user access to applications and enforce authentication policies.

- Scalability and Flexibility:
  - Design a scalable architecture capable of accommodating growing user bases and evolving business requirements.

#### Potential Industries Benefiting from Azure:

- Enterprise Solutions:
  - Streamline access to business applications and enhance security for employees within organizations.
- Education Sector:
  - Provide seamless access to educational resources and online learning platforms for students and faculty members.
- Healthcare Industry:
  - Improve access control and compliance with regulatory requirements for healthcare applications and patient portals.
- Financial Services:
  - Enhance security and user experience for banking and financial applications accessed by clients and employees.
- Government Agencies:
  - Strengthen authentication mechanisms and access controls for government services and portals.
- E-commerce Platforms:
  - Simplify user authentication and enhance security for online shopping platforms and customer accounts.

By integrating Azure SSO with Java backend services and React JS frontend applications, we aim to deliver a scalable, secure, and user-friendly authentication solution that meets the needs of various industries and organizational environments.

## What Needs to Be Done?

### Development Tasks:

#### Azure Setup:

- Create an Azure Active Directory (Azure AD) tenant or use an existing one.
- Register the Java backend and React JS frontend applications in Azure AD.
- Configure required permissions and API permissions for both applications.

#### Java Backend Implementation:

- Develop a Spring Boot application to serve as the backend service.
- Configure Azure AD as the authentication provider using Azure AD Spring Boot Starter.
- Implement authentication endpoints for login and logout.
- Secure backend APIs using Azure AD authentication.

#### React JS Frontend Development:

- Set up a React JS application to serve as the frontend.
- Integrate Microsoft Authentication Library (MSAL) for JavaScript to enable authentication with Azure AD.
- Implement login and logout components for user authentication.
- Access secured backend APIs using acquired tokens.

#### Integration and Testing:

- Integrate the React JS frontend with the Java backend to enable communication.
- Test the SSO flow by logging in from the frontend application.
- Verify token acquisition and authorization for backend API access.
- Ensure smooth redirection and user experience during authentication.

## Documentation and Deployment:

### Documentation:

- Document setup instructions for Azure AD, backend service, and frontend application.
- Provide detailed implementation guidelines for each development task.
- Include code snippets, screenshots, and diagrams to aid understanding and implementation.

### User Documentation:

- Prepare user documentation for seamless adoption of the SSO-enabled applications.
- Provide instructions for users on how to log in, log out, and navigate through the applications.

### Deployment:

- Deploy the applications on a staging environment for validation.
- Document deployment procedures and configurations.
- Monitor performance and address any issues during deployment.

By completing these development tasks and documentation, we aim to successfully integrate Single Sign-On with Azure using Java and React JS, providing a secure and seamless authentication experience for our applications.

Github: <https://github.com/Brilworks-Software/SSO-Azure-POC/>