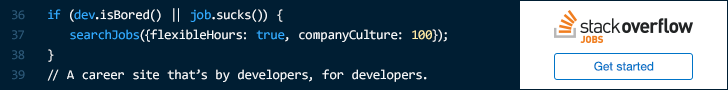
[Azure AD Login/logout implementation for Spring cloud microservices](https://stackoverflow.com/questions/40055187/azure-ad-login-logout-implementation-for-spring-cloud-microservices)

[Ask Question](https://stackoverflow.com/questions/ask)

[](https://engine.adzerk.net/r?e=_dXRtX3NvdXJjZT13ZWJzaXRlJnV0bV9tZWRpdW09YmFubmVyJnV0bV9jb250ZW50PWxlYWRlcmJvYXJkXzYmdXRtX2NhbXBhaWduPWhvdXNlX2Fkc19ob3VzZV9hZHNfUk9TX1NPIn0&s=pAxKg8KoX_LZsl5FAUC5gM3eD5o)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| up vote2down vote[favorite](https://stackoverflow.com/questions/40055187/azure-ad-login-logout-implementation-for-spring-cloud-microservices) | I want to implement login and logout functionality and retrive user details like username and user role using Azure Active Directory.  We are using Docker to deploy [Spring cloud microservices project](https://github.com/sunilsoni/supplier-collaboration) on Azure cloud. Could you please suggest me steps to get user details?  Do we need to secure all microservices edge points using Spring cloud OAuth2 security using JWT or just we can secure one web microservice ? Do I need any permission ,specific user roles to implement this?  [azure](https://stackoverflow.com/questions/tagged/azure) [azure-active-directory](https://stackoverflow.com/questions/tagged/azure-active-directory) [spring-security-oauth2](https://stackoverflow.com/questions/tagged/spring-security-oauth2)   |  |  |  | | --- | --- | --- | | [share](https://stackoverflow.com/q/40055187)[improve this question](https://stackoverflow.com/posts/40055187/edit) | [edited Oct 16 '16 at 14:28](https://stackoverflow.com/posts/40055187/revisions) | asked Oct 15 '16 at 5:09  [[https://i.stack.imgur.com/34u3H.png?s=32&g=1](https://stackoverflow.com/users/1954837/sunil)](https://stackoverflow.com/users/1954837/sunil)  [Sunil](https://stackoverflow.com/users/1954837/sunil)  **122**212 | |
|  | add a comment |

1 Answer

[active](https://stackoverflow.com/questions/40055187/azure-ad-login-logout-implementation-for-spring-cloud-microservices?answertab=active#tab-top)[oldest](https://stackoverflow.com/questions/40055187/azure-ad-login-logout-implementation-for-spring-cloud-microservices?answertab=oldest#tab-top)[votes](https://stackoverflow.com/questions/40055187/azure-ad-login-logout-implementation-for-spring-cloud-microservices?answertab=votes#tab-top)

|  |  |
| --- | --- |
| up vote1down vote | You can find Azure's documentation about OAuth 2.0 support for AAD here  <https://docs.microsoft.com/en-us/azure/active-directory/active-directory-protocols-oauth-code>  I've got an application that's using OAuth 2.0 with a different Authentication Server, and I'm about to see if I can use AAD as the Authentication Server. But, whatever ends up being your Auth Server, the rest of the application should be the same...   * The Auth Server handles the log in (typically as a Single-Sign On pattern) * The Auth Server will return a Json Web Token (at some point, depending on the Grant Type being used to retrieve it) * The JWT should be included in each subsequent request to ensure the caller has authorization   From a Spring perspective, you'll need at least a SSO Client (denoted by the @EnableOAuthSSO annotation). If everything in hosted by that process, you'll need that JWT to call subsequent methods. If you have processes hosted in other processes, it's likely you'll want them secured as well. Using the @EnableResourceServer annotation will configure Spring Security to look for the JWT, just not attempt to retrieve one if the request does not have it.  Unless the endpoint is meant to be publicly accessible, you will want to secure it. Of course, I really don't know the context of your application, so this statement is purely an uninformed opinion based on zero knowledge of what you're trying to do with your application. Take it for what it's worth.  EDIT  This has become a little more complex than I originally thought. I have been able to write some code to dynamically retrieve the public key from Microsoft in order to validate the returned JWT.  But, the main issue is the fact the Azure AD supports Open Id Connect when acting as an Identity/Authentication Server. And, at the moment, spring-security-oauth2 doesn't support Open Id Connect.  I was able to make some small changes to the spring code, but [I did ask the question to the Spring group](https://github.com/spring-projects/spring-security-oauth/issues/909) and they are actively working on adding support for Open Id Connect. They hope to have a release two months (ish?).  For the short term, the oauth2 support doesn't support Open Id Connect. Given this is the protocol used by AAD, the current version of oauth2 won't work with AAD. That said, I will be happy to wait for the official support which shouldn't be too long. |