# SSO Okta valve internals

## code walkthrough:

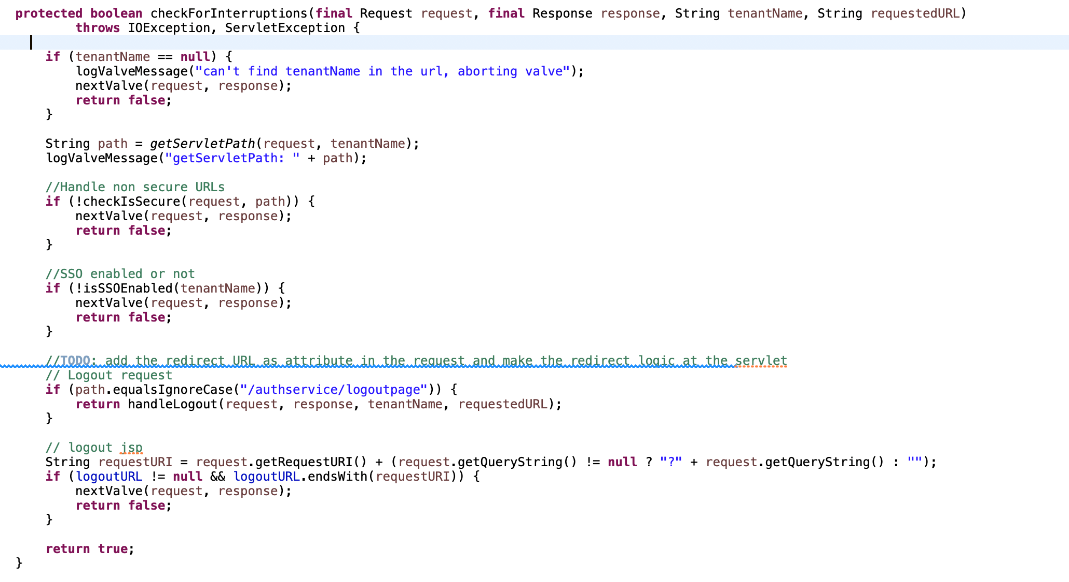
Class OktaValve extends BaseValve which extends ValveBase from catalina

the method that gets invoked in the Valve when a request is made is invoke()

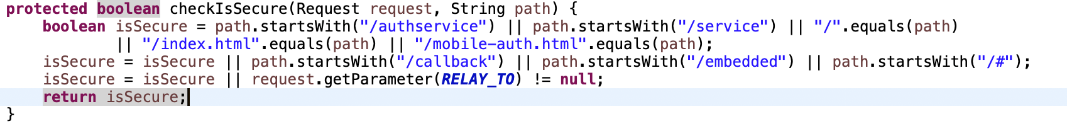


first it extract the URL which the request is sent to in incorta and then extracts the tenant name attached with the URL and converts it to lowercase

then checkForInterruptions() method is called which if returned true then it calls processValve() which is an abstract method implemented in all classes implementing the BaseValve class

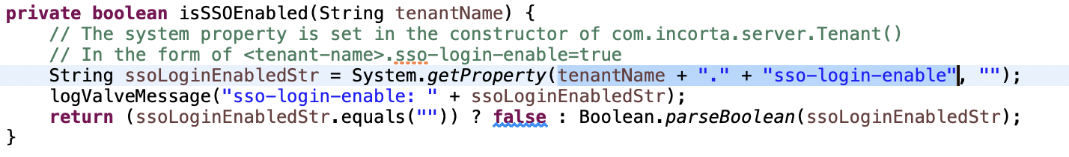


checkForInterruptions() method checks if the the path is secure checkIsSecure() by checking if the path starts with /authservice , /service , /embedded , /# , /callback or equals /, /index.html, /mobile-auth.html or the request has a parameter relayTo



the checks if the SSO is enabled for that tenant by checking if the JVM have property set called <tenantName>.sso-login-enable which its value is set during the server startup

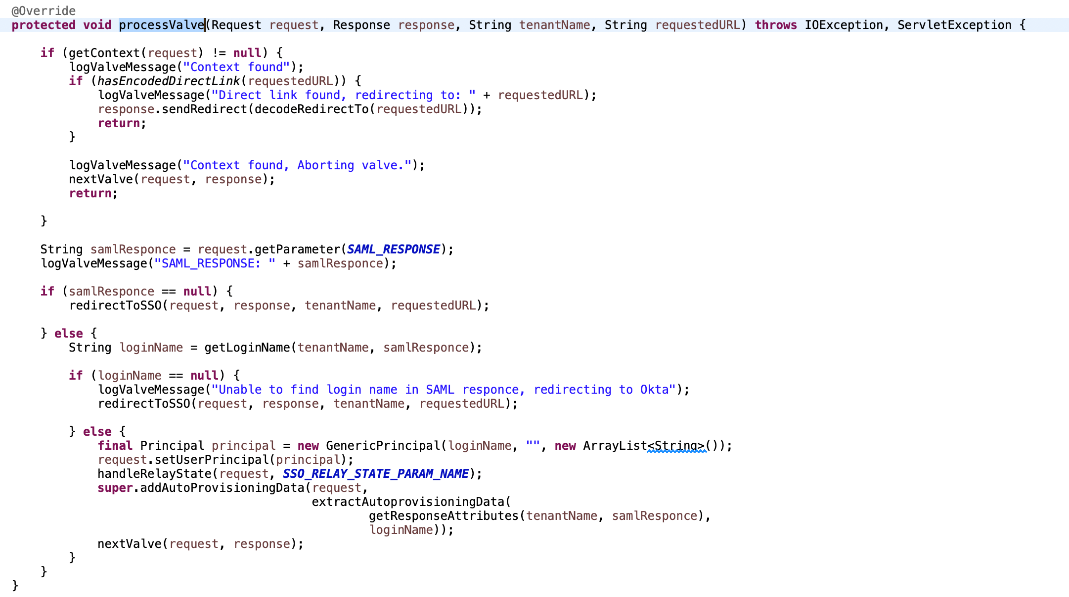
Note that in OktaValve we don’t have access to Incorta classes hence we can’t get the value of the column sso-enabled from the tenant table from the database



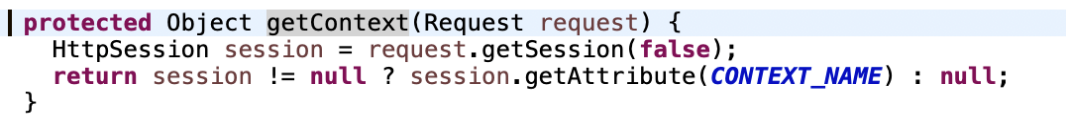
in Tenant class constructor the system property key <tenantName>.sso-login-enable is set to be read later in the Valves



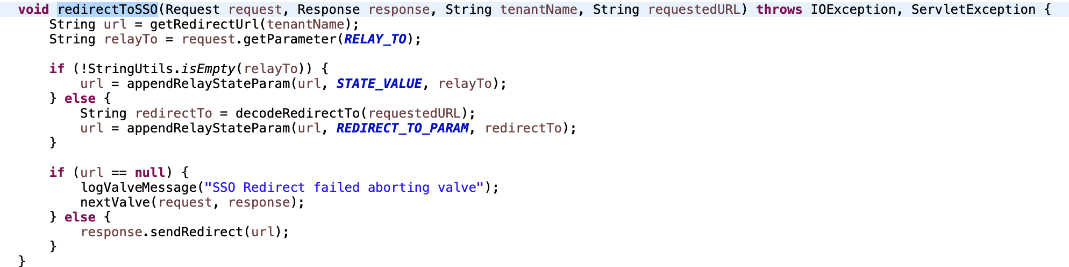
processValve method implementation in OktaValve



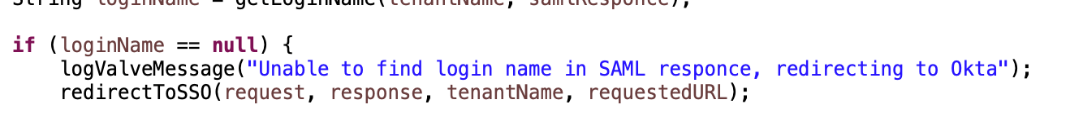
1. it checks if there is a session or not using function getContext



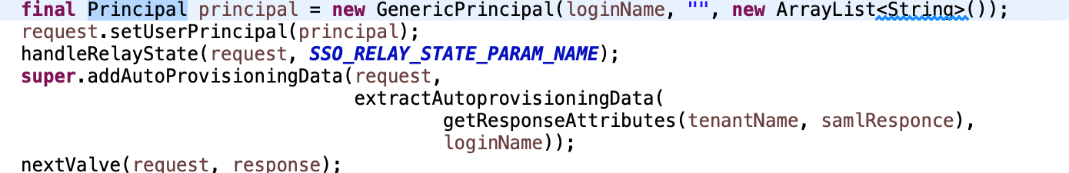
1. assume that the user hasn’t logged in yet so the User doesn’t have session yet
2. checks if there is a parameter in the request called SAMLResponse in case the user hasn’t logged in yet in Okta the it shall be null
3. redirectToSSO method is called, it gets the URL to redirect to Okta and adds the relayState to it



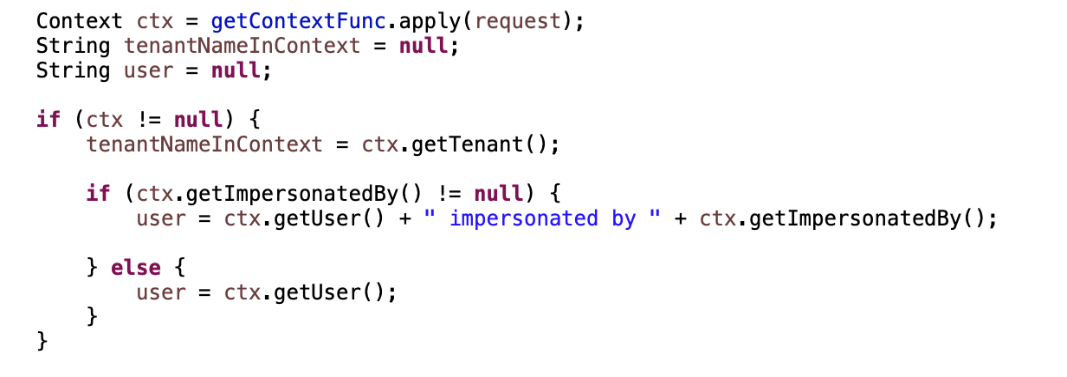
1. after the user gets redirected to Okta and authenticate in Okta, the user gets redirected back to incorta and the request should contain SAMLResponse which contains the user LoginName sent from Okta
2. if the SAMLResponse doesn’t contain LoginName then there is a problem and shall redirect again to SSO (Okta)



1. if the SAMLResponse contains LoginName then create a Principal which is a user defined for tomcat then process the next valve.



1. after all values gets invoked then doFilter function inAuthFilter gets invoked which checks if there is session exists for the current user



1. if the user doesn’t have Session checks if the tenant in URL has SSO enabled then calls loginSSO function which calls AuthServlet.loginSSO which calls AuthServlet.login and creates session for the user

