SSO connection service

Summary

SSO connection service provides the interface with which third party SSO solution can be utilized for the sharing of certificate between independent sites.

Description

The interface between SSO agent and eGovFrame certification system is defined when constructing the certificate service by utilizing the 3rd party SSO solution. The 3rd party SSO solution enables the use of SSO agent function for the sharing of certificate at eGovFrame through the interface implementation.

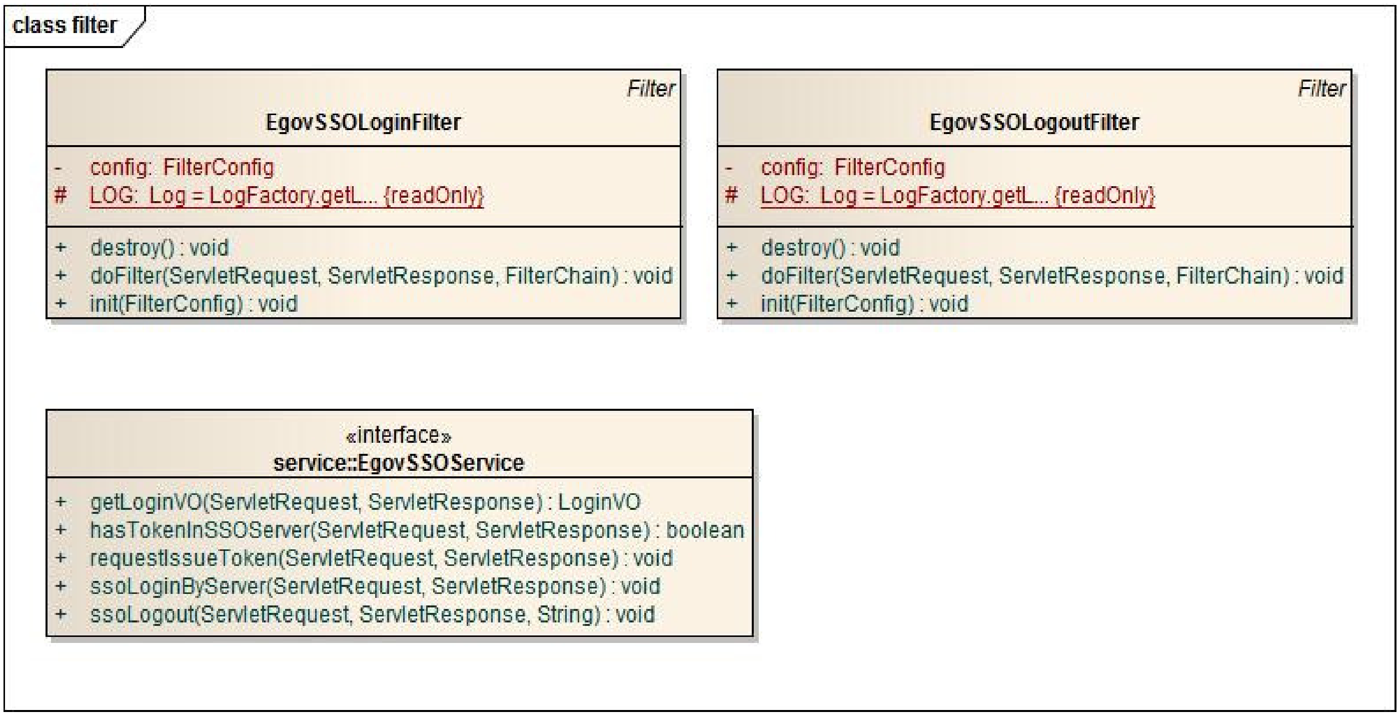
Package Dependency

SSO connection service has direct functional dependency only for the common package (cmm) of element technology and general login package (uat.uia), and it is distributed by being included in the general logon package (uat.uia).

Related Sources

Type	Name of Corresponded Source	Remarks
Service	egovframework.com.uat.uia.sso.service.EgovSSOService.java	The class which defines the SSO connection service interface
Filter	egovframework.com.uat.uia.sso.filter.EgovSSOLoginFilter.java	The filter class which executes SSO certificate through the connection with SSO server
Filter	egovframework.com.uat.uia.sso.filter.EgovSSOLogoutFilter.java	The filter class which processes the global logout of SSO server when the logout is requested

Class Diagram


class filter
Filter
EgovSSOLoginFilter
- config: FilterConfig
# LOG: Log = LogFactory.getL... {readOnly}
+ destroy() : void
+ doFilter(ServletRequest, ServletResponse, FilterChain) : void
+ init(FilterConfig) : void
Filter
EgovSSOLogoutFilter
- config: FilterConfig
# LOG: Log = LogFactory.getL... {readOnly}
+ destroy() : void
+ doFilter(ServletRequest, ServletResponse, FilterChain) : void
+ init(FilterConfig) : void
«interface»
service::EgovSSOService
+ getLoginVO(ServletRequest, ServletResponse) : LoginVO
+ hasTokenInSSOServer(ServletRequest, ServletResponse) : boolean
+ requestIssueToken(ServletRequest, ServletResponse) : void
+ ssoLoginByServer(ServletRequest, ServletResponse) : void
+ ssoLogout(ServletRequest, ServletResponse, String) : void

Implementation example

This is the example of connecting the SSO solution which is used for the national representative integrated certification system by utilizing common component SSO connection system and the certification system of common component, and the detailed implementation example is as follows.

example.zip

- Implement the method which checks whether the certified token is existed at integrated certification server.

```
          /**
           * The method which confirms whether certification is existed at the SSO integrated
certification server
           *
           */
          public boolean hasTokenInSSOServer(ServletRequest request,
                              ServletResponse response) {
                    SSORspData rspData = ssoService.ssoGetLoginData((HttpServletRequest)request);
                    String uid = rspData.getUID();

                    if(uid == null || uid.equals("")){
                              return false;
                    }else{
                              return true;
                    }
          }
```

- Implement the method which requests the issuing of new token to the integrated certification server.

```
         /**
           * The method which requests the generation of certification token to the SSO integrated
certification server
           *
           */
          public void requestIssueToken(ServletRequest request, ServletResponse response) throws
Exception {

                    String serverIp = "";
          String userIp = "";
          String rtrURL = "";
          String clientPort = "";

                    serverIp = InetAddress.getLocalHost().getHostAddress();
                    userIp = EgovClntInfo.getClntIP((HttpServletRequest)request);
                    clientPort = ":" + request.getServerPort();
          rtrURL = ((HttpServletRequest)request).getRequestURI();


          LoginVO loginVO =
(LoginVO)((HttpServletRequest)request).getSession().getAttribute("loginVO");

                    ssoService.ssoReqIssueToken((HttpServletRequest)request, // Servlet request object
                                   (HttpServletResponse)response,// Servlet reply object
                  "form-based",                           // Certification method
                  loginVO.getUniqId(),                                        // Unique ID
                  loginVO.getUserSe(),                              // ID identification information
                  "",                 // SSN
                  "http://" + serverIp + clientPort + rtrURL, // return url
                  userIp,             // client ip
                  serverIp);                    // agent ip
          }
```

- Implement the method which performs the local certification with certified token at integrated certification server.

```
/**
 * The method which processes local login by utilizing the token of certification server when the certification is existed at the SSO integrated certification server
 *
 */
public void ssoLoginByServer(ServletRequest request,
                ServletResponse response) throws Exception {
        SSORspData rspData = ssoService.ssoGetLoginData((HttpServletRequest)request);

        LoginVO loginVO = new LoginVO();
        loginVO.setUniqId(rspData.getUID());
        loginVO.setUserSe(rspData.getCN());

        // Preparation of local login
        loginVO = loginService.actionLoginByEsntlId(loginVO);


        //((HttpServletRequest)request).getSession().setAttribute("uid", rspData.getUID());

        // Spring security login

((HttpServletResponse)response).sendRedirect("/j_spring_security_check?j_username=" + loginVO.getUserSe() + loginVO.getId() + "&j_password=" + loginVO.getUniqId());

//((HttpServletRequest)request).getRequestDispatcher("/j_spring_security_check?j_username=" + loginVO.getUserSe() + loginVO.getId() + "&j_password=" + loginVO.getUniqId()).forward(request, response);


}
```

- Implement the method which generates the LoginVO object with the basis of token information of integrated certification server.

```
/**
 * The method which generates the loginVO object with the basis of token information
 *
 */
public LoginVO getLoginVO(ServletRequest request, ServletResponse response){
        SSORspData rspData = ssoService.ssoGetLoginData((HttpServletRequest)request);

        LoginVO loginVO = new LoginVO();
        loginVO.setUniqId(rspData.getUID());
        loginVO.setUserSe(rspData.getCN());

        return  loginVO;
}
```

- Implement the method which requests global logout (token deletion) to the integrated certification server.

```
/**
 * Process the Global Logout which deletes the SSO certification information.
 * returnURL : The URL address to be returned after the Global Logout
 * @throws IOException
 *
 */
public void ssoLogout(ServletRequest request, ServletResponse response, String returnURL) throws IOException{
```

```
            ((HttpServletResponse)response).sendRedirect("/exam/sso/globalLogout.do?returnURL=" +
returnURL);
```