3DI APPLET FIRMA DIGITALE

Versione: 4.0
Rilascio: 28/06/2017
Compatibilità con tutti i browser in seguito a conversione alla tecnologia JNLP.

Requisiti client

Per utilizzare l'applet di firma è necessario che sul client sia correttamente installato e collegato un dispositivo di firma digitale.

Inoltre è necessario che sia installata una JRE che attraverso la funzionalità Java Web Start eseguirà l'applet.

Integrazione

Per integrare l'applet in un'applicazione web è necessario seguire i seguenti passaggi:

1. Creare una directory nella webapp accessibile dall'esterno e copiarvi le cartelle e il loro contenuto
    1. jars
    2. dependencies
    3. jnlp

1. Includere nel template dell'applicativo da cui si vuole lanciare l'applet la sezione html necessaria.
    1. un esempio si può trovare nel file javaWebStartTest.html (vedi sezione tecnica successiva)
    2. bisogna implementare il metodo closeFirma() per effettuare il reload del documento.

Integrazione - Aspetto Tecnico

Per avviare la generazione a RUN-TIME del file JNLP che consentirà l'avvio dell'applet di firma è necessario invocare la jsp **3disign.jsp** passando i vari parametri nella queryString.

La costruzione dell'url è completamente automatizzata da uno script Javascript, bisogna solo effettuare il binding dei vari parametri presenti nelle input hidden con i valori del bean.

```
<div>
            <p>
                Per firmare i documenti è necessario scaricare ed eseguire il file jnlp.
                Al termine del processo premere sul pulsante 'chiudi' per ricaricare il documento.
            </p>

            <button onclick="return closeFirma();">Chiudi</button>

            <a id="startApplet" style="display: none;"
href="http://localhost:9080/3diSignatureAppletServlet/jnlp/3disign.jsp"></a>

            <input type="hidden" name="href" value="jnlp/3disign.jsp" />
            <input type="hidden" name="code"
value="it.tredi.applet.signature.SignatureApplet.class" />
            <input type="hidden" name="archive" value="3disignapplet.jar" />
            <input type="hidden" name="jnlp_href"
value="http://10.17.61.155:9080/3diSignatureAppletServlet/jnlp/3disign.jsp" />
            <input type="hidden" name="DownloadFilesServiceURL"
value="http://10.17.61.155:9080/3diSignatureAppletServlet/servlet" />
            <input type="hidden" name="InputFile"
value="http://10.17.61.155:9080/3diSignatureAppletServlet/tests/eXtraWayWebServices.pdf;http://10.
17.61.155:9080/3diSignatureAppletServlet/tests/eXtraWayWebServices2.pdf" />
            <input type="hidden" name="FileNameURL"
value="FileNameURL=eXtraWayWebServicesS.pdf;eXtraWayWebServices2S.pdf" />
             <input type="hidden" name="OutputFileType" value="p7m" />
            <input type="hidden" name="SendToServerURL"
value="http://10.17.61.155:9080/3diSignatureAppletServlet/servlet?action=upload;http://10.17.61.15
5:9080/3diSignatureAppletServlet/servlet?action=upload" />
            <input type="hidden" id="3diSignatureAppletParamCookies" name="cookies" value="" />
            <input type="hidden" name="InputOutputFilesSeparator" value=";" />

            <script type="text/javascript">
                /* gestire eventualmente i cookie dal document e non passati alle input da bean*/
                //document.querySelector('input[id="3diSignatureAppletParamCookies"]').value =
document.cookie;
```

```
                /* JSF vuole &amp;amp; come separatore dei campi nella queryString dell'URL */
                var paramSeparator = "&";

                /* generalmente solo i parametri InputFile e SendToServerURL sono complessi da
richiedere l'encoding
                    tuttavia dipende da come vengono passati dal bean
                */
                var href = document.querySelector('#startApplet').href;
                href += "?" +
                        "code="+
document.querySelector('input[name="code"]').value + paramSeparator +
                         "archive="+
document.querySelector('input[name="archive"]').value + paramSeparator +
                        "jnlp_href="+
document.querySelector('input[name="jnlp_href"]').value + paramSeparator +
                        "DownloadFilesServiceURL="+
document.querySelector('input[name="DownloadFilesServiceURL"]').value + paramSeparator +
                        "InputFile="+
encodeURIComponent(document.querySelector('input[name="InputFile"]').value) + paramSeparator +
                        "FileNameURL="+
document.querySelector('input[name="FileNameURL"]').value + paramSeparator +
                        "OutputFileType="+
document.querySelector('input[name="OutputFileType"]').value + paramSeparator +
                        "SendToServerURL="+
encodeURIComponent(document.querySelector('input[name="SendToServerURL"]').value) + paramSeparator
+
                        "cookies="+
document.querySelector('input[name="cookies"]').value + paramSeparator +
                        //"debug="+                          "true" + paramSeparator +
                        "InputOutputFilesSeparator="+
document.querySelector('input[name="InputOutputFilesSeparator"]').value;

                document.querySelector('#startApplet').href = href;

                /* decommentare per far partire il download del file in automatico al caricamento
della pagina*/
                //startFirma();

                function startFirma(){
                    var href = document.querySelector("#startApplet").href;
                    window.location.href = href;
                }

                /* implementare il proprio meccanismo di reload del document a fine processo di
firma
                    questa funzione viene richiamata dal tasto chiudi in alto*/
                function closeFirma() {
                    alert("TODO implement reload document")
                    return false;
                }
            </script>
        </div>
```

Installazione dispositivi

Durante la fase di installazione viene copiato un modulo (.dll su windows) che si occupa dell'interfacciamanto tra il dispositivo e i suoi driver. E' necessario verificare che tale modulo venga copiato correttamente e che non vada in conflitto con altre versioni precedenti.

Le utlime smartcard (Aruba, InfoCert, ...) stanno distribuendo l'ultima implementazione del modulo **bit4xpki.dll**. Tale modulo al momento sembra risolvere tutte le problematiche di compatibilità rilevate con i vari dispositivi “precedenti” che invece utilizzavano diverse implementazioni dei moduli *bit4opki.dll* e *bit4ipki.dll*.

Risoluzione problematiche comuni

Applicazione bloccata per motivi di sicurezza

I jar dell'applet non sono firmati correttamente o hanno un certificato non valido. Se l'errore si presente in un ambiente di produzione è necessario rinnovare il certificato di firma della 3D e rieffettuare il signer dei jar. (esiste uno script nel progetto).

Il processo di firma si blocca al 10% perché non rileva il dispositivo di firma

Il dispositivo non viene letto perchè la porta USB non da alimentazione. Dovrebbe essere sufficiente scollegare e ricollegare il dispositivo e premere RIPROVA.

Il processo di firma si blocca al 80%

Problema dovuto alla non corretta comunicazione tra dispositivo e driver. Nei dispositivi che utilizzavano i driver bit4ipki.dll e bit4opki.dll il problema era molto frequente.

Attraverso una serie di analisi era emerso che con l'installazione di Dike 4.5.3 veniva installata una versione molto robusta di tale driver. La maggior parte dei problemi si sono risolt installando Dike sul client.

Con la nuova versione del driver bit4xpki.dll il problema non si è mai presentato, inoltre tale driver è perfettamente retrocompatibile con i vecchi dispositivi.