GOBIERNO DE ESPAÑA
MINISTERIO DE ASUNTOS ECONÓMICOS Y TRANSFORMACIÓN DIGITAL

FNC
Asistido por AutenticA

Aplicación de ejemplo FNC

Documentación técnica

Versión: 1.0.0

Fecha de la versión: 01/12/2022

Madrid, 01 de diciembre de 2022

Elaborado por la Secretaría General de Administración Digital

© Ministerio de Asuntos Económicos y Transformación Digital

ÍNDICE

1. OBJETO 3

2. PARÁMETROS DE CONFIGURACIÓN 4

2.1. Fichero web.xml 4

2.2. Ficheros apache 5

2.3. Ficheros autenticaConfig.dat 5

3. FLUJO DE APLICACIÓN DE EJEMPLO 6

3.1. Acceso a la aplicación 6

3.2. Firma del documento 8

3.2.1. Autenticación previa 8

3.2.1. Firma 9

3.3. Descarga de ficheros generados 10

1. OBJETO

El objeto del presente documento es proporcionar a integradores y desarrolladores información técnica sobre la aplicación de ejemplo de integración con FNC, de modo que pueda entenderse y modificarse fácilmente el código de ejemplo para su reutilización en la integración de aplicaciones externas.

Para cada funcionalidad, el documento incluye una descripción del funcionamiento de las pantallas y las clases de la aplicación, así como el flujo de datos entre la aplicación integrada y el servidor de FNC. También se incluye una descripción de los parámetros configurables en la aplicación de ejemplo.

2. PARÁMETROS DE CONFIGURACIÓN

2.1. Fichero web.xml

En el desplegable ***AutenticaFncDemo.war*** se podrá configurar el fichero ***web.xml*** para adaptarlo a las necesidades de la aplicación que se integra. A continuación, se muestran los parámetros configurables y los valores por defecto:

- urlEnvironment: entorno para realizar la firma no criptográfica. Por defecto, se realiza la firma en el entorno de servicios estables de Autentica (se-autentica.redsara.es), pero se podrá configurar para utilizar otros entornos de Autentica.
- appPades: id de Autentica de la aplicación invocante, configurada para firma con formato Pades. Por defecto, se utiliza la aplicación de prueba con identificador 7223, pero se podrá modificar con el id de la aplicación que se quiere integrar con FNC.
- appXades: id de Autentica de la aplicación invocante, configurada para firma con formato Pades. Por defecto, se utiliza la aplicación de prueba con identificador 7243, pero se podrá modificar con el id de la aplicación que se quiere integrar con FNC.

```
<init-param>
  <param-name>urlEnvironment</param-name>
  <param-value>https://se-autentica.redsara.es</param-value>
</init-param>
<!-- Aplicación de ejemplo configurada con Pades -->
<init-param>
  <param-name>appPades</param-name>
  <param-value>7223</param-value>
</init-param>
<!-- Aplicación de ejemplo configurada con Xades -->
<init-param>
  <param-name>appXades</param-name>
  <param-value>7243</param-value>
</init-param>
```

Dado que FNC – Autentica permite únicamente configurar cada aplicación con un tipo de firma (Pades o Xades), la aplicación de ejemplo incluye dos identificadores de aplicación diferentes, una por cada tipo de firma. En una integración real únicamente será preciso emplear un id de aplicación.

2.2. Ficheros apache

Se deberá hacer uso de un nuevo dominio para la autenticación de la firma no criptográfica. Se modificarán los ficheros ***autentica_80.conf*** y ***autentica_443.conf***, añadiendo un host virtual con la configuración actual de Autentica. El nuevo dominio se llamará ***se-autentica-fnc.redsara.es***.

```
<VirtualHost *:80>
    ServerName se-autentica-fnc.redsara.es
    Header edit Set-Cookie "(?i)^((?:(?!;s?HttpOnly).)+)$" "$1; HttpOnly"
```

En los mismos se añadirán las líneas:

```
JkMount /Autentica* autentica_tomcat851
JkMount /cas342Dir4* autentica_tomcat851
JkMount /samlAutentica* autentica_tomcat851
```

2.3. Ficheros autenticaConfig.dat

Se deberá configurar la url de autenticación para la firma no criptográfica para que apunte al nuevo dominio de firma. Se modificará el fichero ***autenticaConfig.dat*** y se añaden los parámetros ***fncCasEnv*** y ***fncSamlEnv*** con las rutas correspondientes al nuevo dominio.

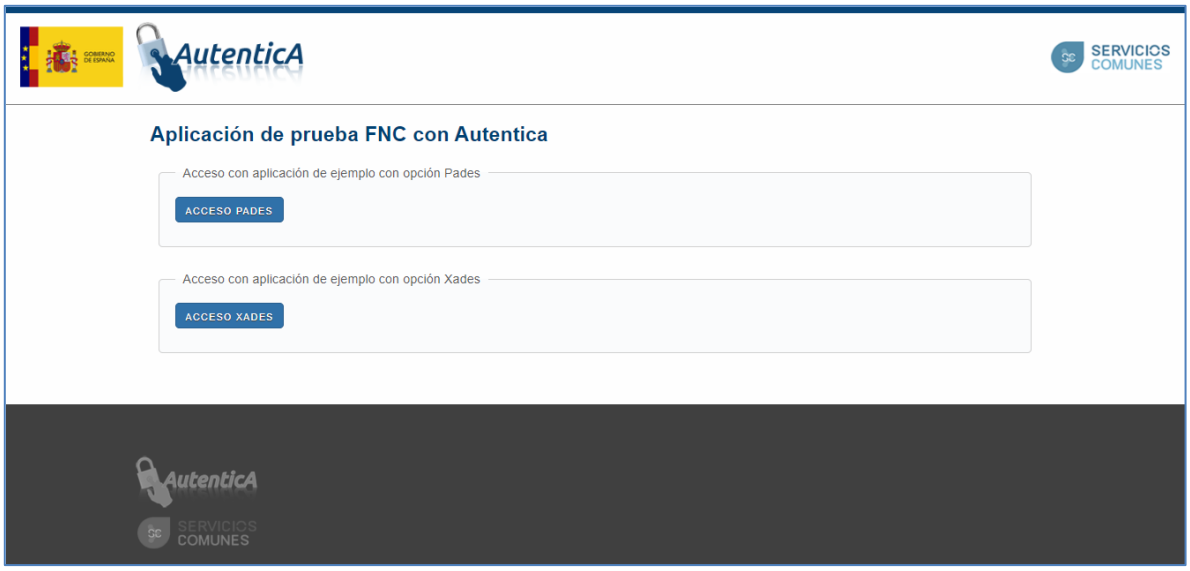
```
fncCasEnv=https://se-autentica-fnc.redsara.es/cas342Dir4
fncSamlEnv=https://se-autentica-fnc.redsara.es/samlAutentica
```

3. FLUJO DE APLICACIÓN DE EJEMPLO

3.1. Acceso a la aplicación

Al acceder a la ruta https://se-autentica.redsara.es/AutenticaFncDemo se envía la acción ***goToIndex*** en la clase ***AutenticaFncDemoHandler*** que llevará a la pantalla principal de la aplicación de ejemplo.

```
public String goToIndex(HttpServletRequest req, HttpServletResponse res) {
    log.debug("AutenticaFncDemoHandler.goToIndex.INI");
    return "OK";
}
```



En la misma, se mostrarán 2 funcionalidades disponibles, una para la firma no criptográfica con opción Pades y otra para la FNC con opción Xades. En dicha pantalla se cargan los identificadores de aplicaciones configuradas en el fichero web.xml del ejecutable para cada una de las funcionalidades. Además, se incluye la extensión correspondiente a cada formato de firma.

```
<fieldset>
    <legend>Acceso con aplicaci&oacute;n de ejemplo con opci&oacute;n Pades</legend>
    <form id="fFncAutPades" name="fFncAutPades" action="../servlet/AutenticaFncDemoServlet?action=goToAuthenticationFnc&type=JSP" method="post">
        <input type="submit" id="submitAutenticaPades" value="Acceso Pades" />
        <input type="hidden" id="action" name="action" value="goToAuthenticationFnc" />
        <input type="hidden" id="appId" name="appId" value="<%=StaticData.getAPP_PADES()%>"/>
        <input type="hidden" id="ext" name="ext" value=".pdf"/>
    </form>
</fieldset>
<fieldset>
    <legend>Acceso con aplicaci&oacute;n de ejemplo con opci&oacute;n Xades</legend>
    <form id="fFncAutXades" name="fFncAutXades" action="../servlet/AutenticaFncDemoServlet?action=goToAuthenticationFnc&type=JSP" method="post">
        <input type="submit" id="submitAutenticaPades" value="Acceso Xades" />
        <input type="hidden" id="action" name="action" value="goToAuthenticationFnc" />
        <input type="hidden" id="type" name="type" value="JSP" />
        <input type="hidden" id="appId" name="appId" value="<%=StaticData.getAPP_XADES()%>"/>
        <input type="hidden" id="ext" name="ext" value=".xml"/>
    </form>
</fieldset>
```

Desde la pantalla principal se podrá acceder a la aplicación de prueba configurada en el fichero web.xml, mediante el botón de acceso correspondiente a cada uno de los formatos Pades o Xades. Una vez se pulsa en el botón, se envía la acción ***goToAuthenticationFnc*** en la clase

***AutenticaFncDemoHandler*.** En dicha acción de recibe el identificador de la aplicación y se realiza la llamada a Autentica para autenticar al usuario.

```
public String goToAuthenticationFnc(HttpServletRequest req, HttpServletResponse res) {
    try {
        String sAppId = req.getParameter("appId");
        req.getSession().setAttribute("appId", sAppId);
        log.debug("AutenticaFncDemoHandler.goToAuthenticationFnc.sAppId:" + sAppId);

        String sExt = req.getParameter("ext");
        req.getSession().setAttribute("ext", sExt);
        log.debug("AutenticaFncDemoHandler.goToAuthenticationFnc.sExt:" + sExt);

        // Se realiza la llamada a autentica para validación del usuario
        String sRedirect = StaticData.getURL_ENV()
                + "/Autentica/servlet/AutenticaServlet?action=goToAutentica&appId=" + sAppId;
        log.debug("invokeGoToAutenticaFnc.sRedirect:" + sRedirect);
        res.sendRedirect(sRedirect);

        return "OK";

    } catch (Exception e) {
        log.error("AutenticaFncDemoHandler.goToAuthenticationFnc.Error: " + e.toString());
        return "KO";
    }
}
```

Autentica retornará el control a la aplicación de ejemplo AutenticaFncDemo enviando la acción ***goToAutenticaFncDemo*,** la cual llevará a la pantalla de selección de documento para la firma.

```
public String goToAutenticaFncDemo(HttpServletRequest req, HttpServletResponse res) {
    log.debug("AutenticaFncDemoHandler.goToAutenticaFncDemo.INI");

    return "OK";
}
```



En esta pantalla se cargará el identificador de aplicaciones recibido tras la autenticación.

```
<fieldset>
    <legend>Acceso a FNC</legend>
    <form id="fFncAuthen" name="fFncAuthen"
        action="../servlet/AutenticaFncDemoServlet?action=goToAutenticaFnc&type=JSP"
        enctype="multipart/form-data" method="post">
        <label>Seleccionar documento a firmar</label> <input type="file"
            id="datafile" name="datafile" style="width: 100%" /> <br /> <input
            type="submit" id="submitAutenticaPades" value="Firmar documento" />
        <input type="hidden" id="action" name="action"
            value="goToAutenticaFnc" /> <input type="hidden" id="appId"
            name="appId" value="<%=sAppId%>" />
    </form>
</fieldset>
```

3.2. Firma del documento

Una vez el usuario se encuentra autenticado, podrá seleccionar el documento y enviarlo para ser firmado. Para ello, se sigue el siguiente flujo de la aplicación.

3.2.1. Autenticación previa

Se enviará la acción ***goToAutenticaFnc*** en la clase ***AutenticaFncDemoHandler,*** donde se obtiene el contenido del fichero cargado y el identificador de la aplicación. Con los datos obtenidos se realiza una llamada a Autentica para la autenticación previa de firma no criptográfica.

```
// Obtenemos el fichero cargado y el id de la aplicación
UploadEngine uploadFile = new UploadEngine(req, res);
String[] sFileData = uploadFile.uploaderDocument();
```

```
String sAppId = sFileData[5];
// Realiza autenticación a la firma y se obtiene el hash
String sRedirect = StaticData.getURL_ENV()
        + "/Autentica/servlet/AutenticaServlet?action=goToAutenticaFnc&appId=" + sAppId;
log.debug("invokeGoToAutenticaFnc.sRedirect:" + sRedirect);
res.sendRedirect(sRedirect);
```

Una vez realizada la autenticación, Autentica retornará el control a la aplicación de ejemplo enviando la acción ***goToXmlFncDocument,*** donde se recibe el hash generado, el contenido, nombre y extensión del fichero.

```
String sHash = req.getParameter("FNC_HASH");
log.debug("AutenticaFncDemoHandler.goToXmlFncDocument.FNC_HASH: " + sHash);

String sHashXML = req.getParameter("FNC_XML");
log.debug("AutenticaFncDemoHandler.goToXmlFncDocument.FNC_XML: " + sHashXML);

String[] sHashValues = new UtilityXML().getHashValue(sHashXML);
log.debug("AutenticaFncDemoHandler.goToXmlFncDocument.sHashValues[0]: " + sHashValues[0]);
log.debug("AutenticaFncDemoHandler.goToXmlFncDocument.sHashValues[1]: " + sHashValues[1]);
log.debug("AutenticaFncDemoHandler.goToXmlFncDocument.sHashValues[2]: " + sHashValues[2]);

String sFileContent = (String) req.getSession().getAttribute("fncFileContent");
log.debug("AutenticaFncDemoHandler.goToXmlFncDocument.sFileContent= " + sFileContent);
String sFileName = (String) req.getSession().getAttribute("fncFileName");
log.debug("AutenticaFncDemoHandler.goToXmlFncDocument.sFileName= " + sFileName);
String sFileExtension = (String) req.getSession().getAttribute("fncFileExtension");
log.debug("AutenticaFncDemoHandler.goToXmlFncDocument.sFileExtension= " + sFileExtension);
```

Se crea un xml con los datos obtenidos y se envía a la firma.

```
String sFileXML = "<?xml version='1.0' encoding='UTF-8'?>" + "<fnc>" + "<fileName>" + sFileName
        + "</fileName>" + "<file>" + sFileContent + "</file>" + "<ext>" + sFileExtension + "</ext>"
        + "<userName>" + sHashValues[1] + "</userName>" + "<appId>" + sHashValues[2] + "</appId>" + "<hash>"
        + sHashValues[0] + "</hash>" + "<csv></csv>" + "</fnc>";

log.debug("AutenticaFncDemoHandler.goToXmlFncDocument.sFileXML " + sFileXML);

req.setAttribute("hash", sHashValues[0]);
req.setAttribute("userName", sHashValues[1]);
req.setAttribute("appId", sHashValues[2]);
req.setAttribute("sFileXML", sFileXML);
```

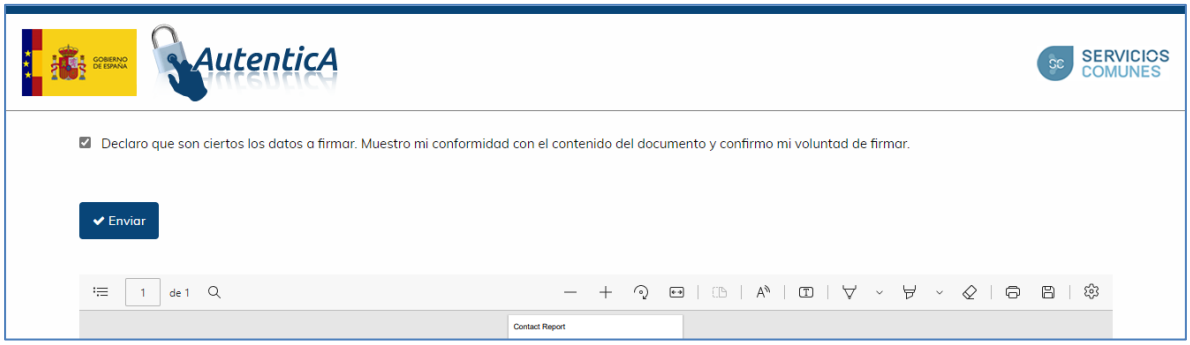
3.2.1. Firma

Se realizará la llamada a la firma no criptográfica enviando la acción ***goToXMLFncDocument*** y como parámetros el hash, el usuario y el identificador de la aplicación, obtenidos de la autenticación previa, así como el xml generado con dichos datos.

```
<form id="fFncAut" name="fFncAut"
    action="<%=StaticData.getURL_ENV() %>/Fnc/servlet/FncServlet"
    method="post">
    <input type="hidden" id="hash" name="hash" value="<%=sHash %>" />
    <input type="hidden" id="userName" name="userName" value="<%=userName %>" />
    <input type="hidden" id="appId" name="appId" value="<%=appId %>" />
    <input type="hidden" id="sFileXML" name="sFileXML" value="<%=sFileXml%>"/>

    <input type="hidden" id="action" name="action" value="goToXMLFncDocument"/>
    <input type="hidden" id="type" name="type" value="JSP"/>
</form>
```

La firma no criptográfica llevará a la pantalla de confirmación de voluntad de firmar, donde se mostrará el documento a firmar y una opción a marcar para declarar la conformidad con la firma del documento.



Se firmará el documento y se retornará el control a la aplicación de ejemplo enviando la acción ***goToFncResponse***. En esta acción se recibirá el contenido del fichero firmado y el contenido del justificante firmado, en ambos casos codificado; además, se recibirá el CSV generado en la firma. Con dichos datos se llevará a la pantalla de datos de respuesta donde se mostrará el contenido del fichero y el CSV.

```
String sFileData = req.getParameter("fileData");
log.debug("AutenticaFncDemoHandler.goToFncResponse.sFileData[Parameter]:" + sFileData);
String sCSV = req.getParameter("csv");
log.debug("AutenticaFncDemoHandler.goToFncResponse.sCSV[parameter]: " + sCSV);
String sJustifData = req.getParameter("justifData");
log.debug("AutenticaFncDemoHandler.goToFncResponse.sJustifData[Parameter]:" + sJustifData);

req.setAttribute("fileData", sFileData);
req.setAttribute("justifData", sJustifData);
req.setAttribute("csv", sCSV);
```



3.3. Descarga de ficheros generados

Desde la pantalla de respuesta se podrá descargar el fichero y el justificante, mediante los enlaces habilitados.

```
<div class="download">
 <a href="#" onclick="javascript:fileDownload('fichero')">Descargar fichero</a>
    <br/>
    <a href="#" onclick="javascript:fileDownload('justificante')" >Descargar justificante</a>
</div>
```

Para la descarga de ambos, se enviará la acción ***openAttachment*** en la clase ***ManagementFilesHandler***, donde se recibirá el nombre del documento a descargar, la extensión y el contenido del documento.

```
String sFileName = (String) request.getParameter("fileName");
String sFileExt = (String) request.getParameter("fileExt");
```

```
if ("justificante".equalsIgnoreCase(sFileName)) {
    sFileData = (String) request.getParameter("justifData");
    sFileDownloadName = "Justificante.pdf";

    if (sFileData == null)
        sFileData = (String) request.getAttribute("justifData");
    if (sFileData == null)
        sFileData = (String) request.getSession().getAttribute("justifData");

} else {
    sFileData = (String) request.getParameter("fileData");
    sFileDownloadName = "Fichero" + sFileExt;

    if (sFileData == null)
        sFileData = (String) request.getAttribute("fileData");
    if (sFileData == null)
        sFileData = (String) request.getSession().getAttribute("fileData");
}
```

Con los datos recibidos se genera el fichero con el contenido del documento firmado y se podrá realizar la descarga.