Table of Contents

Der Up- und Download von Dateien ist in Applikationen kaum wegzudenken, egal ob es sich dabei um den Download von Reports oder den Upload von Importdaten bzw. Dokumenten handelt. In allen Fällen möchte man vor und/oder nach der Aktion weitere Aktionen starten bzw. auf die Daten zugreifen.

Von JVx wird der Up- und Download von Dateien bereits in der Basis unterstützt und ermöglicht eine sehr komfortable und vor allem einfache Verwendung.

Anwendungsbeispiel

Unsere Applikation verwaltet Firmen und zu jeder Firma kann ein Logo hinterlegt werden (Upload). Auf Wunsch kann das Logo für Werbezwecke gespeichert werden (Download).

Für den Up- und Download wird je ein Button bereitgestellt. Dieser kann abhängig von den Benutzerrechten akt-/deaktiviert werden.

Wir zeigen die Verwendung durch Client und Server Actions und beginnen mit einem Auszug aus der Client Implementierng für den Upload:

```
/**
 * Selects the company logo with a file chooser.
 *
 * @throws Throwable if an error occurs while file choosing
 */
public void doUploadLogo() throws Throwable
{
    getApplication().getLauncher().getFileHandle(this,
                                                 "uploadLogo", "Choose
company logo");
}

/**
 * Sends the company logo to the server.
 *
 * @param pFileHandle the company logo
 * @throws Throwable if a communication error occurs
 */
public void uploadLogo(IFileHandle pFileHandle) throws Throwable
{
    getConnection().callAction("uploadLogo",
                               rdbCompany.getValue("CPNY_ID"),
                               pFileHandle);
}
```

Die Client Implementierung für den Download:

```
/**
 * Downloads the company logo and opens a save dialog.
 *
 * @throws Throwable if a communication error occurs
 */
```

```
public void doDownloadLogo() throws Throwable
{
    IFileHandle fileHandle =
(IFileHandle)getConnection().callAction("downloadLogo",
rdbCompany.getValue("CPNY_ID"));

    if (fileHandle != null)
    {
        getApplication().getLauncher().saveFileHandle(fileHandle);
    }
}
```

Die Server Implementierung für den Upload:

```
/**
 * Stores the company logo.
 *
 * @param pCpnyId the company id
 * @param pLogo the company file handle
 * @throws Exception if an error occurs during store
 */
public void uploadLogo(BigDecimal pCpnyId, IFileHandle pLogo) throws
Exception
{
    CallableStatement statement = null;

    try
    {
        //database mode
        statement = getDataSource().getConnection().prepareCall(...);
        statement.setBigDecimal(1, pCpnyId);
        statement.setString(2, pFileHandle.getFileName());
        statement.setBinaryStream(3,
                                  new
BufferedInputStream(pFileHandle.getInputStream()),
                                  (int)pFileHandle.getLength());
        statement.execute();
    }
    finally
    {
       try
       {
            statement.close();
       }
       catch (Throwable pThrowable)
       {
           // Do nothing!
       }
    }
}
```

Die Server Implementierung für den Download:

```
/**
 * Sends the company logo to the client.
 *
 * @param pCpnyId the company id from which we want the logo
 * @return the company file handle or <code>null</code> if the company is
not available
 * @throws Exception if an error occurs when accessing the company
 */
public RemoteFileHandle downloadFile(BigDecimal pCpnyId) throws Exception
{
   CallableStatement statement = null;

   try
   {
      statement = getDataSource().getConnection().prepareCall(...);
      statement.setBigDecimal(1, pScjaId);
      statement.registerOutParameter(2, Types.VARCHAR);
      statement.registerOutParameter(3, Types.BLOB);

      statement.execute();

      String fileName = statement.getString(2);
      if (fileName == null)
      {
         return null;
      }
      else
      {
         return new RemoteFileHandle(fileName,
statement.getBlob(3).getBinaryStream());
      }
   }
   finally
   {
      try
      {
          statement.close();
      }
      catch (Throwable pThrowable)
      {
          // Do nothing!
      }
   }
}
```

From:
https://doc.sibvisions.com/ - **Documentation**

Permanent link:
https://doc.sibvisions.com/de/jvx/client/gui/upload_download

Last update: **2018/01/31 21:30**