

Table of Contents

If you need a CSV export of a databook, there's already a built-in solution in JVx. Of course, it depends on your databook implementation, whether it's a RemoteDataBook or a MemDataBook, or completely different.

But usually, the following snippet will help:

```
IFileHandle file;

if (pDataBook instanceof RemoteDataBook)
{
    pDataBook.saveAllRows();

    AbstractConnection pConnection =
    ((RemoteDataBook)pDataBook).getDataSource().
        getConnection();

    file = (IFileHandle)pConnection.call(pDataBook.getName(), "createCSV",
        columnNames, labels, filter,
        pDataBook.getSort(),
LocaleUtil.getDefault().toString());
}
else
{
    ByteArrayOutputStream bos = new ByteArrayOutputStream();

    DataBookCSVExporter.writeCSV(pDataBook, bos,
        columnNames, labels, filter, null);

    bos.close();

    file = new
FileHandle(DataBookCSVExporter.formatCSVFileName(pDataBook.getName()),
        bos.toByteArray());
}

ILauncher launcher = ApplicationUtil.getLauncher(pComponent);

if (pShow)
{
    launcher.showFileHandle(file);
}
else
{
    launcher.saveFileHandle(file, null);
}
```

You'll need a databook (pDataBook) and the launcher or a UI component (pComponent). The export of remote databooks is implemented in the remote storage, so simply call the remote method. The export of memory databooks will be done with DataBookCSVExporter.

If you want to create CSV exports of server-side storages, have a look at following project:

[AESStorageExport](#).

The project is an extension that allows CSV creation of one or more storages and it supports ZIP creation - with or without password protection.

From:

<http://doc.sibvisions.com/> - **Documentation**

Permanent link:

http://doc.sibvisions.com/jvx/common/util/csv_export



Last update: **2020/06/15 12:59**