

Table of Contents

Durch die [Server Actions](#) können beliebige Funktionen aus der Business Logik vom Client angesprochen werden. Im Normalfall handelt es sich dabei um Aktionen die nicht sehr Zeitintensiv sind wie z.B. Eingabevalidierungen.

Aus diesem Grund arbeitet die Kommunikation zwischen Client und Server synchron, sprich der Client stellt eine Anfrage an den Server und wartet bis die Antwort vom Server zurückgeliefert wird. Während der Wartezeit wird der Mauscursor in der Applikation als Sanduhr dargestellt, und der Anwender kann keine weiteren Aktionen durchführen.

Wenn z.B ein Bericht erstellt werden soll, der sehr zeitintensive Berechnungen und Datenanalysen durchführt, dann ist eine längere Wartezeit nicht von Vorteil.

Für zeitintensive Aktionen ist eine asynchrone Kommunikation ohne weiteren Aufwand möglich. Das bedeutet aber nicht, daß die komplette Kommunikation asynchron arbeiten muss. Es obliegt dem Applikationsentwickler zu entscheiden welche Aktionen asynchron ausgeführt werden sollen.

Die Übertragung der Ausführungsergebnisse wird vollautomatisch durchgeführt. Der Entwickler muss sich daher um keine Besonderheiten kümmern.

Example

Unsere Applikation ermöglicht die Erstellung von Berichten denen Millionen von Datensätzen zugrunde liegen. Diese Datensätze müssen jedoch erst ausgewertet und verdichtet werden. Die Laufzeit für die Erstellung eines Berichtes liegt bei ca. 3 Minuten.

Der Anwender startet einen Bericht mit einem Button (siehe [Client Actions](#)) und erhält einen Hinweis wenn der Bericht erstellt wurde. In der Zwischenzeit kann wie gewohnt weitergearbeitet werden.

Für die Lösung benötigen wir im ersten Schritt eine Server Action:

```
/**
 * Creates a statistic report.
 *
 * @param pId the master data ID
 * @param pName the report name entered from the user
 */
public String startReport(BigDecimal pId, String pName)
{
    ReportGenerator rgen = new ReportGenerator();

    rgen.createReport(pId);

    return pName;
}
```

Die Server Action wird mit nun asynchron aufgerufen. Das wird durch die Verwendung eines Callback listeners erledigt:

```
/**
 * Starts the creation of the statistic report.
 *
```

```
* @throws Throwable if it's not possible to start report creation
*/
public void doStartReport() throws Throwable
{
    getConnection().callAction(new ICallbackListener()
        {
            public void callback(CallbackEvent
                pEvent)
            {
                try
                {
                    showInformation(Showcase.this,
                        (String)pEvent.getObject());
                }
                catch (Exception e)
                {
                    ExceptionHandler.raise(e);
                }
            }
        },
        "startReport",
        bdId,
        "Report: " + bdId);
}
```

Sobald der Report erstellt wurde wird der Client darüber informiert. In unserer Lösung wird ein Hinweis angezeigt.

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

Permanent link:
https://doc.sibvisions.com/de/jvx/communication/async_callback

Last update: **2018/01/31 21:42**

