



Jasper 4 Oracle Dokumentation

Erstellungsdatum: 29.11.2018
Autoren: Philipp Ostmeyer
letzte Änderung: 26.04.2019 von: Philipp Ostmeyer

TEAM GmbH
Hermann-Löns-Straße 88
33104 Paderborn

1 Inhaltsverzeichnis

1	INHALTSVERZEICHNIS	2
2	HINWEISE FÜR DEN LESER	3
2.1	DOKUMENTENHISTORIE	3
2.2	ANSPRECHPARTNER.....	3
3	INSTALLATION	4
3.1	BENÖTIGTE SOFTWARE	4
3.2	INSTALLATION DER SERVER KOMPONENTE	4
3.3	INSTALLATION DER PL/SQL CLIENT-API	5
3.3.1	<i>Vergabe der nötigen Berechtigungen</i>	<i>5</i>
3.4	INSTALLATION DES LIZENZSCHLÜSSELS	5
4	KONFIGURATION	6
4.1	PORT.....	6
4.2	OBERFLÄCHE.....	6
4.2.1	<i>Administrations-Account.....</i>	<i>6</i>
4.2.2	<i>Datumsformat.....</i>	<i>6</i>
4.3	DEFAULT-DATENBANKVERBINDUNG	6
4.4	LOKALE VERZEICHNISSE	7
4.4.1	<i>Library Verzeichnis</i>	<i>7</i>
4.4.2	<i>Deployment Verzeichnis.....</i>	<i>7</i>
4.5	TREIBER	7
4.6	SMTP-SERVER	8
4.7	NULL-WERT	8
5	BENUTZUNG	8
5.1	BEREITSTELLEN EINES REPORTS.....	8
5.2	AUFRUFEN EINES BEREITGESTELLTEN REPORT	11
5.2.1	<i>Parameterübergabe</i>	<i>11</i>
5.2.2	<i>Reportexportierung.....</i>	<i>12</i>
5.2.2.1	<i>Parameter</i>	<i>12</i>
5.3	DRUCKERANZEIGE	13
5.4	NEULADEN DER REPORTDEFINITIONEN	14
5.5	VERWALTUNGSOBERFLÄCHE	14
5.5.1	<i>Log-Ansicht.....</i>	<i>14</i>
5.5.2	<i>Neuladen von Reports</i>	<i>15</i>
5.6	PL/SQL CLIENT	15
5.6.1	<i>View URL</i>	<i>16</i>
5.6.2	<i>Download URL.....</i>	<i>17</i>
5.6.3	<i>Print Report</i>	<i>17</i>
5.6.4	<i>Mail Report.....</i>	<i>18</i>
5.6.5	<i>Lokales Speichern</i>	<i>18</i>
5.6.6	<i>Neuladen der Reportdefinitionen.....</i>	<i>19</i>
6	VERIFIKATION EINER KORREKTEN INSTALLATION	20

2 Hinweise für den Leser

2.1 Dokumentenhistorie

Datum	Autoren	Bemerkung
29.11.2018	Philipp Ostmeyer	Initiale Version

2.2 Ansprechpartner

Name	Erreichbar über
Philipp Ostmeyer	Tel.: 05254 8008-58 eMail: po@team-pb.de
Wolf G. Beckmann	Tel.: 05254 8008-39 eMail: wb@team-pb.de

3 Installation

3.1 Benötigte Software

Software	Version	Anmerkung
Java Runtime Environment	1.8-u152	oder höher

3.2 Installation der Server Komponente

Entpacken Sie das Jasper4Oracle Archiv in das Installationsverzeichnis Ihrer Wahl.

Für die Installation wird der JDBC-Treiber von Oracle benötigt (Beispielsweise „ojdbc8.jar“). Der Treiber kann entweder aus einer Oracle-Installation(„ORACLE_HOME/jdbc/lib/...“) entnommen werden oder per Download direkt von Oracle bezogen werden.

Zu finden unter dem Link:

<https://www.oracle.com/technetwork/database/application-development/jdbc/downloads/index.html>

Kopieren Sie das „ojdbcXX.jar“-File in das resources/jr_library Verzeichnis.

Je nach Betriebssystem können Sie nun den Server über „startServer“ (Linux) oder „startServer.bat“ (Windows) starten.

Unter Linux muss das Start-Script ausführbar gemacht werden. Dazu führen Sie auf der Kommandozeile folgenden Befehl aus:

```
chmod u+x startServer  
./startServer
```

```
startServer.bat
```

Beim Aufruf dieser Befehle können Funktionalitäten des OS genutzt werden, um den Log des Servers in Dateien umzulenken. Zum Beispiel unter Linux:

```
chmod u+x startServer  
./startServer > startServer.log 2>1
```

3.3 Installation der PL/SQL Client-API

3.3.1 Vergabe der nötigen Berechtigungen

Starten Sie SQL-Plus als „sys“ Datenbankbenutzer und vergeben sie die Rechte zur Ausführung der Packages „utl_http“ und „utl_url“ an das Schema, dem die TEAM_J4O-Package installiert werden soll.

Öffnen Sie einen Terminal im Database Verzeichnis der Installation.

```
sqlplus /nolog
SQL> CONNECT SYS as SYSDBA
Enter password: <SYS_password>
SQL> grant execute on utl_http to <Schema>;
SQL> grant execute on utl_url to <Schema>;
SQL> exit;
```

Installieren Sie das TEAM_J4O-Package und bereiten Sie die Berechtigung für die Verbindung mit dem J4O-Server vor.

Dazu starten Sie erneut SQL-Plus, aber diesmal im benutzenden Schema. Beachten Sie, dass Sie sich im Unterverzeichnis „Database“ des Installationsverzeichnis befinden.

```
sqlplus /nolog
SQL> CONNECT <Schema>
Enter password: <Schema_password>
SQL> @TEAM_J4O.pkh
SQL> @TEAM_J4O.pkb
SQL> @prepareACL
IP or Name of the J4O-Server (eg. localhost): <Server_Name>
Port of the J4O-Server (eg. 8090): <Port_Number>
```

Das „prepareACL.sql“ Script erstellt eine „setup_acl.sql“ Datei. Diese wird genutzt, um die Berechtigung zur Verbindung mit dem J4O-Server zu vergeben. Dazu muss dieses Script unter dem „sys“ Benutzer ausgeführt werden.

```
sqlplus /nolog
SQL> CONNECT SYS as SYSDBA
Enter password: <SYS_password>
SQL> @setup_acl
```

3.4 Installation des Lizenzschlüssels

Um einen Lizenzschlüssel zu aktivieren, muss eine zusätzliche Zeile in der Konfigurationsdatei („Jasper4Oracle.config“) folgendermaßen angefügt werden:

```
LICENSE_KEY=<KEY>
```

4 Konfiguration

4.1 Port

Um den Port des Jasper4Oracle-Servers zu ändern, editieren Sie das jeweils relevante Startscript („startServer.sh“ bzw. „startServer.bat“).

Dazu können Sie einfach den Aufrufparameter „port“ auf den gewünschten Wert setzen (z.B. „-port=8090“).

4.2 Oberfläche

4.2.1 Administrations-Account

Die Daten des Administrations-Accounts werden im „Jasper4Oracle.config“-File definiert.

Um diese Einstellungen zu ändern öffnen Sie dieses mit einem beliebigen Editor und ändern die Zeilen, welche mit „ADMIN_USER“ und „ADMIN_PW“ starten.

4.2.2 Datumsformat

Um das Datumsformat der Darstellung auf Oberfläche des J4O-Servers zu ändern, können Sie die mit „DISPLAY_DATE_FORMAT“ beginnende Zeile ändern. Hier sind alle Datumsformatierungen nach Java Konvention erlaubt. Eine Definition dieser finden Sie unter:

<https://docs.oracle.com/javase/8/docs/api/java/text/SimpleDateFormat.html>

4.3 Default-Datenbankverbindung

Es gibt die Möglichkeit eine Default-Datenbankverbindung einzurichten. Diese wird benutzt, wenn keine valide Datenbankverbindung zur Reportexportierung angegeben wird.

Um diese einzustellen, kann dem „Jasper4Oracle.config“-File folgende drei Zeilen angefügt werden:

```
DEFAULT_DB_URL=<DB_URL>  
DEFAULT_DB_USER=<DB_SCHEMA>  
DEFAULT_DB_PASSWORD=<DB_SCHEMA_PASSWORD>
```

4.4 Lokale Verzeichnisse

4.4.1 Library Verzeichnis

Das Library-Verzeichnis enthält alle Jar-Files, welche zur Laufzeit benötigt werden. Dazu gehören Datenbanktreiber, die JasperLibrary als auch die jeweiligen Archive der bereitgestellten Reports.

Im „Jasper4Oracle.config“-File kann das Library-Verzeichnis konfiguriert werden. Die mit „LIB_DIRECTORY“ beginnende Zeile enthält diese Definition. Standardmäßig zeigt dieses auf das „resources“-Unterverzeichnis des Installationsverzeichnisses.

Alle Pfade werden als Unterverzeichnisse im Startverzeichnis des J4O-Servers gesehen.

Falls absolute Pfade benutzt werden sollen, muss diese Definition mit „/“ (Linux) oder einer Laufwerkbezeichnung, z.B. „C:/“ (Windows) anfangen.

Das Symbol „\“ ist nicht erlaubt und muss durch „/“ ersetzt werden.

4.4.2 Deployment Verzeichnis

In der „Jasper4Oracle.config“-Datei kann das Deployment-Verzeichnis konfiguriert werden, indem die mit „DEPLOYMENT_DIRECTORY“ beginnende Zeile verändert wird.

Indem konfigurierten Verzeichnis, und etwaigen Unterverzeichnissen, können Dateien, die exportiert werden, abgelegt werden.

Wenn Benutzer nicht nur in ein Verzeichnis, sondern auf das gesamte System schreiben dürfen sollen, kann hier „/“ eingetragen werden.

Bestehende Dateien werden durch den J4O-Server nie überschreiben.

4.5 Treiber

Für den J4O-Server kann ein Treiber konfiguriert werden, mit welchem die Datenbankverbindung, welche zur Reportsgeneration benutzt wird, aufgebaut werden soll.

Um einen anderen Treiber zu benutzen muss dessen Jarfile in das „resources/jr_library“ Verzeichnis abgelegt werden.

Des Weiteren, müssen zwei Konfigurationsparameter in der „Jasper4Oracle.config“-Datei geändert werden:

Jeder JDBC-Treiber wird durch eine zu ladende Klasse und einen Connector-Typen definiert.

Diese müssen in den Zeilen „JDBC_DRIVER_NAME“ und „DB_CON_DRIVER_PREFIX“ angegeben werden. Die jeweiligen speziellen Konfigurationen können beim Hersteller des Treibers nachgelesen werden.

4.6 SMTP-Server

Um einen SMTP-Server zu konfigurieren werden folgende Parameter benötigt. Diese Parameter stehen jeweils in ihrer eigenen Zeile in der Konfigurationsdatei. Sollten sie nicht vorhanden sein, können sie einfach unten angefügt werden:

„MAIL_SMTP_HOST“ – IP oder DNS-Name des anzusprechenden SMTP-Servers
„MAIL_SMTP_PORT“ – Port des anzusprechenden SMTP-Servers

„MAIL_SMTP_USER“ – Nutzer, welcher zur Authentifikation mit dem Server genutzt wird.

„MAIL_SMTP_PW“ – Das zugehörige Passwort.

Wenn die Parameter „MAIL_SMTP_USER“ und „MAIL_SMTP_PW“ leer sind, wird versucht eine Verbindung ohne Authentifikation durchzuführen.

„MAIL_SMTP_FROM_ADDRESS“ – Default Mail-Adresse, welche genutzt wird, wenn keine andere angegeben wird.

4.7 NULL-Wert

Da die Kommunikation mit dem J4O-Server über http erfolgt, muss die Parametrisierung der Reports über Strings erfolgen. Um ein explizites NULL an einen Report zu übergeben, muss ein NULL-String konfiguriert werden.

Dazu kann in der Konfigurationsdatei eine Zeile mit dem Schlüssel

„STRING_PARAMETER_NULL_VALUE=“ angefügt werden.

Dahinter kann dann der gewünschte String angegeben werden, welcher immer als NULL interpretiert wird.

5 Benutzung

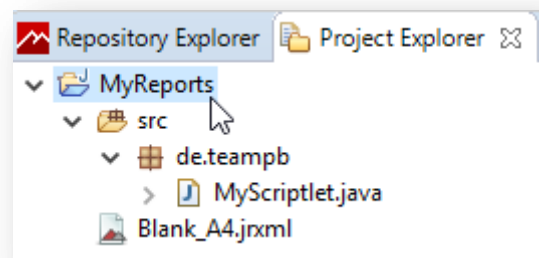
5.1 Bereitstellen eines Reports

Ein Report ist immer durch ein „Jar“-Archive gegeben. Dieses enthält verschiedenste Komponenten:

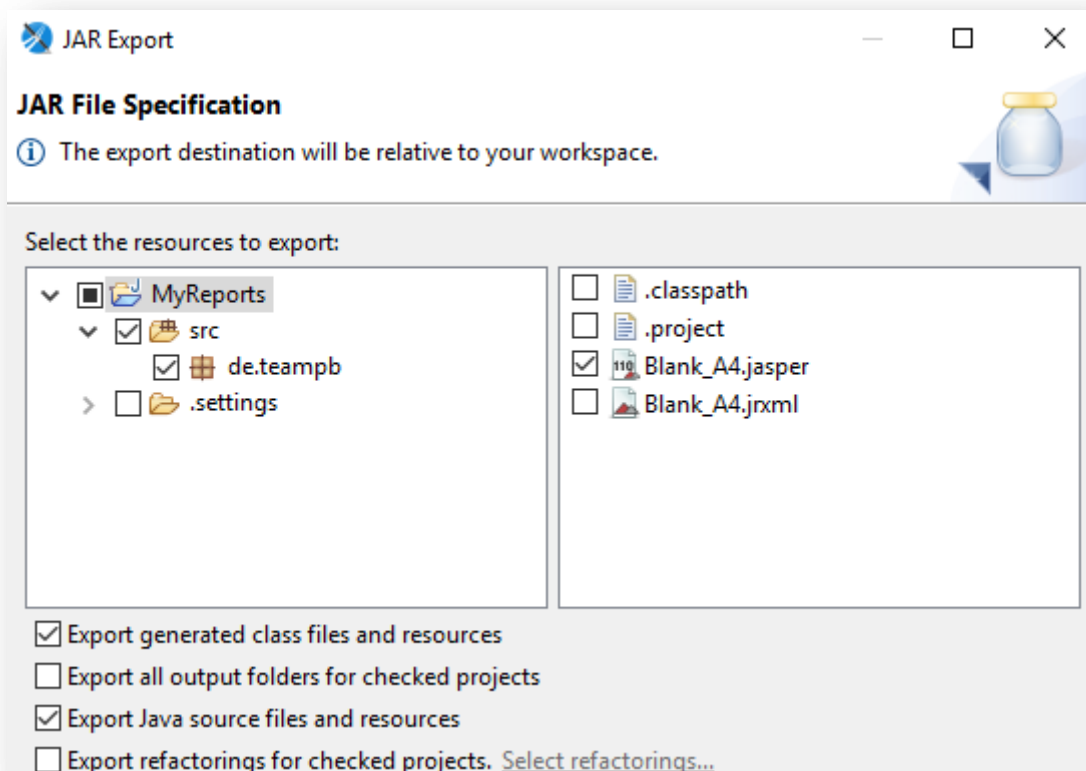
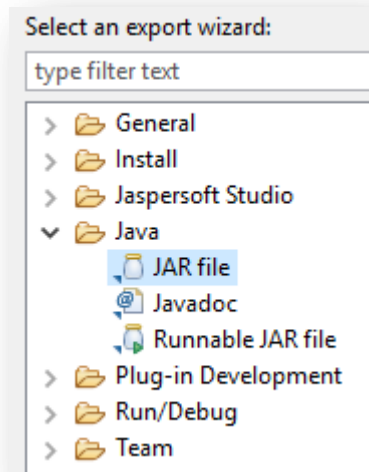
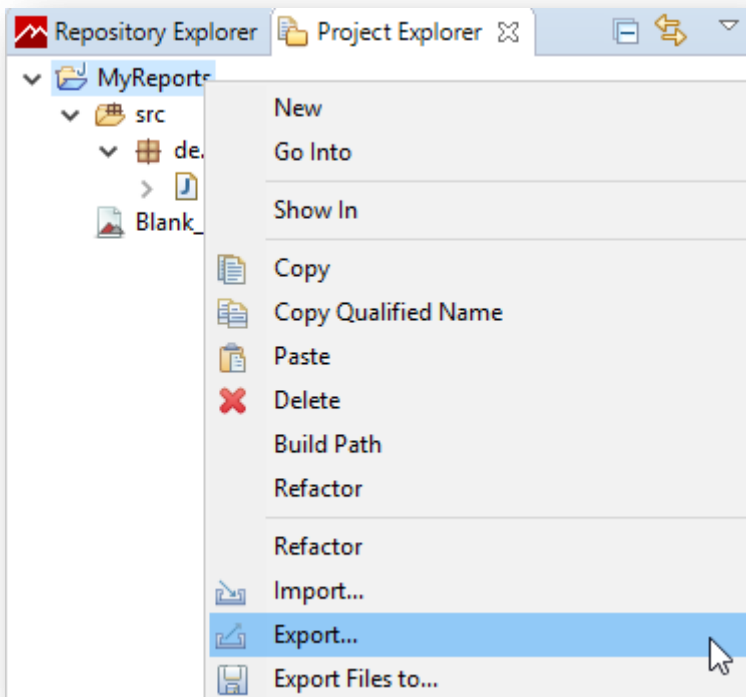
Jasperreports-Dateien	Diese Dateien sind durch die Endung „.jasper“ gekennzeichnet und werden vom JasperReports Studio aus den „jrxml“-Dateien erstellt. Sollte im JasperReports Studio eine Ordnerstruktur innerhalb eines Projekts verwendet werden, so muss diese Struktur auch im Jar beibehalten werden.
Java-Dateien	Diese Dateien sind durch die Endung „.class“ gekennzeichnet und werden vom JasperReports Studio aus den „java“-Dateien kompiliert. Da Java-Klassen in einem Package sein sollten muss

	dieses durch die Ordner Struktur abgebildet sein. Das JasperReports Studio erstellt beim Anlegen einer Klasse oder eines Packages diese Struktur. Des Weiteren, wird diese Struktur bei Generation der „.class“-Files beibehalten (siehe „bin“-Ordner des Projekts).
Resource-Dateien	Unter Ressourcen versteht man allgemein alle anderen Dateien, welche zur Reportgenerierung benötigt werden. Hauptsächlich sind damit Bild-Dateien gemeint, welche in die Reports eingebettet werden sollen. Bei diesen Dateien sollte genauso verfahren werden wie mit Jasperreports-Dateien. D.h. etwaige Ordnerstrukturen im Projekt sollten beibehalten werden.

Ein solches Jarfile kann über den Jasperreports Studio-internen Exporter gebaut werden:



I Rechtsklick auf das jeweilige Projekt



Ein so erstelltes Jar kann dann auf dem Server bereitgestellt werden.
Dazu kann es in eines von zwei Verzeichnissen gelegt werden:

reports	Dies ist das Verzeichnis in dem Reports abgelegt werden, welche nicht von anderen Reports referenziert werden. D.h. es gibt keine Einschränkungen bzgl. der Reports.
global_libraries	Dies ist das Verzeichnis in dem Report abgelegt werden, welche von anderen Reports referenziert werden. Hier gelten folgende Einschränkungen: Es dürfen keine zwei gleichnamige „jasper“-Dateien innerhalb aller Jar-Archive in diesem Verzeichnis existieren. Es dürfen keine zwei gleichnamige „class“-Dateien, welche im gleichen Package liegen, innerhalb aller Jar-Archive in diesem Verzeichnis existieren.

Die Referenz von Reports anderer Projekte wird über die Projekt-/Source-Folder-Referenz der Projekte erstellt.
Dieses ermöglicht die Reports so zu Referenzieren, als ob sie im gleichen Projekt lägen. Diese Referenz wird auf dem J4O-Server automatisch vorgenommen und muss nicht weiter konfiguriert werden.

Um ein neues oder bestehendes Jar aktiv zu schalten muss ein Neuladevorgang gestartet werden. Dies kann über die Oberfläche geschehen (siehe TBD) oder über eine HTTP-Schnittstelle (siehe 5.4).

5.2 Aufrufen eines bereitgestellten Report

Ein Aufruf eines bereitgestellten Reports gliedert sich in die folgenden zwei http-Aufrufe:

5.2.1 Parameterübergabe

Die Parameterübergabe geschieht über einen HTTP-Aufruf. Dieser ist wie folgt definiert:

HTTP-Method	POST
URL	<IP>:<PORT>/jasper4oracle/reporting/createToken
Content-Type	„application/x-www-form-urlencoded“
Parameter	Werden im Body des Aufrufs mitgegeben.

Die Aufrufparameter eines Reports gliedern sich in zwei Kategorien:

Reportparameter werden genutzt, um die vom Jasperreport definierten Inputparameter zu füllen. Diese werden einfach im Aufruf mitübergaben.
D.h. die Namen der Parameter müssen nicht verändert werden und werden direkt an den Report weitergereicht.

J4O-Parameter werden benutzt, um Metaparameter zur Ausführung zu definieren. Darunter fallen z.B. Reportnamen oder Datenbankdefinitionen.

Die Folgenden Parameter sind definiert:

Parameter	Beschreibung	Benötigt
j4o_report	Name des auszuführenden Jar-Archivs	X
j4o_mainJasper	Name des im Jar-Archive befindlichen, auszuführenden „jasper“-Files	X
j4o_databaseUrl	DNS-Name oder IP der Datenbank, an welche eine Verbindung für den Report hergestellt werden soll	(X)
j4o_dbUser	Name des Schemas/Datenbankbenutzers, mit welchem der Report ausgeführt werden soll	(X)
j4o_dbPassword	Passwort des benutzten Accounts	(X)
j4o_format	Das zu erzeugende Format. Muss eines der folgenden beinhalten: PDF, DOCX, HTML, PRINT, CSV	
j4o_locale	Die zu benutzende Sprachumgebung. Die Definition dieser kann unter folgendem Link nachgeschlagen werden: https://docs.oracle.com/javase/7/docs/api/java/util/Locale.html#forLanguageTag(java.lang.String)	

Bei erfolgreichem Aufruf wird ein Token zurückgegeben. Dieser symbolisiert das Recht einen Report einmal mit den vorher definierten Parametern aufzurufen.

5.2.2 Reportexportierung

Zur Reportexportierung wird ein in 5.2.1 erzeugter Token benötigt. Ist dieser vorhanden, kann über einen HTTP-Aufruf der Exportvorgang gestartet werden. Dieser Aufruf ist wie folgt definiert:

HTTP-Method	GET
URL	<IP>:<PORT>/jasper4oracle/reporting/export
Parameter	Werden im Header des Aufrufs mitgegeben.

Die folgenden Parameter sind für den Exportvorgang zulässig:

5.2.2.1 Parameter

5.2.2.1.1 „token“

In diesen Parameter wird der zu exportierende Token übergeben. Der Parameter ist immer benötigt.

5.2.2.1.2 „dest_type“

Dieser Parameter beschreibt wie der Report nach dem Exportvorgang zur Verfügung gestellt wird.

Einer der folgenden Werte werden erwartet:

VIEW - Dem öffnenden Browser wird mitgeteilt, dass der Report (wenn möglich) im aufrufenden Tab angezeigt werden soll. Dieser Wert wird genutzt, wenn kein anderer angegeben wird.

DOWNLOAD - Dem öffnenden Browser wird mitgeteilt, dass der Report in einem Downloaddialogfenster geöffnet wird. Dieser ermöglicht dem Benutzer einen Speicherort zu wählen (falls kein Standardordner eingestellt ist)

LOCAL_FILE - Der Report wird auf der Maschine, auf welcher der J4O-Server gestartet ist, gespeichert.
Der Parameter „destination“ muss gesetzt werden. Dieser wird als Ordnerangabe interpretiert, in welchem der Report abgelegt wird. Diese Angabe wird relativ zum Konfigurationsparameter *4.4.2 Deployment Verzeichnis* interpretiert.
Der Parameter „file_name“ muss gesetzt werden. Ist dieser mit der korrekten Dateiendung versehen, wird er so genutzt wie definiert.
Andernfalls wird diese hinzugefügt.

PRINT - Der Report wird direkt an einen Drucker geschickt. Dazu muss als erstes der gewünschte Drucker auf der Maschine, auf welcher der J4O-Server gestartet ist, installiert werden. Dessen Name kann dann im Parameter „destination“ mitgegeben werden, um den Report dort zu drucken.
Der Parameter „destination“ ist immer mitzugeben, wenn „dest_type“ auf „PRINT“ gesetzt ist.
Eine Abfrage der installierten Drucker ist möglich(vgl. 5.3).

5.2.2.1.3 „destination“

Dieser Parameter wird von den Reportbereitstellungstypen LOCAL_FILE und PRINT benutzt. Dort kann die jeweilige Benutzung nachgelesen werden.

5.2.2.1.4 „file_name“

Dieser Parameter wird von den Reportbereitstellungstypen LOCAL_FILE benutzt. Dort kann die Benutzung nachgelesen werden.

5.2.2.1.5 „print_copies“

Dieser Parameter wird von den Reportbereitstellungstypen PRINT benutzt. Dort kann die Benutzung nachgelesen werden.

5.3 Druckeranzeige

Der J4O bietet eine HTTP-Schnittstelle an, um die auf der Maschine verfügbaren Drucker anzuzeigen. Diese ist wie folgt definiert:

HTTP-Method	GET
URL	<IP>:<PORT>/jasper4oracle/admin/printers

Diese Schnittstelle ist maschinenlesbar.

5.4 Neuladen der Reportdefinitionen

Um bereitgestellte Jardateien aktiv zu schalten, müssen diese neu geladen werden. Dazu kann die folgende HTTP-Schnittstelle genutzt werden:

HTTP-Method	GET
URL	<IP>:<PORT>/jasper4oracle/admin/redeploy

Die folgenden Parameter sind hierfür zulässig:

Parameter	Beschreibung	Benötigt
level	Gibt an, welche Reports aktiv geschaltet werden sollen. Folgende Werte sind erlaubt: „single“ – Ein einzelner Report soll behandelt werden. Hierzu wird der Parameter „name“ benötigt. „all“ – Alle Dateien im Verzeichnis „deployment/reports“ werden behandelt. „global“ – Alle Dateien in den Verzeichnissen „deployment/reports“ und „deployment/global_libraries“ werden behandelt.	X
name	Name des Jar-Archives, das aktiviert werden soll.	(X)

5.5 Verwaltungsoberfläche

Die graphische Oberfläche des J4O-Servers ist unter folgender HTTP-Schnittstellen zu erreichen:


HTTP-Method	GET
URL	<IP>:<PORT>/jasper4oracle/ui/log.xhtml

Die Anmeldedaten werden in der Jasper4Oracle.config definiert (siehe 4.2.1). Die Oberfläche ist für den Browser „Google Chrome“ optimiert, kann aber auch mit aktuellen Versionen des „Mozilla Firefox“, „Microsoft Internet Explorer“ und „Microsoft Edge“ benutzt werden.

Nach der Anmeldung stehen folgende Dienste zur Verfügung:

5.5.1 Log-Ansicht

In der Log-Ansicht wird der Status aller aktuellen Reportprozesse gezeigt.

Über das Symbol  können diese Stati auf einen Zeitraum eingeschränkt werden.

Über das Symbol  kann eine Detailansicht geöffnet werden. Diese beinhaltet Angaben zu Aufrufparametern und eine Liste von Logeinträgen.

5.5.2 Neuladen von Reports

Die Oberfläche bietet die Möglichkeit, alle Reports neuzuladen, indem der Button „Redeploy“ gedrückt wird.

Dies ist äquivalent zum Aufruf in 5.4 mit dem Parameter „level“ auf „global“ gesetzt.

5.6 PL/SQL Client

Mit dem Jasper4Oracle-Server wird ein PL/SQL-Package zur Verfügung gestellt, welches als Client-Schnittstelle fungiert. Eine Anleitung zur Installation ist unter 3.3 zu finden.

Der PL/SQL-Client bedient die oben beschriebenen HTTP-Schnittstellen, d.h. sie erwartet dieselben Parameter.

Diese Parameter gliedern sich in die zwei Gruppen der allgemeinen und report-spezifischen Parameter:

Allgemeine Parameter gelten für jeden Aufruf und können einmal pro Session gesetzt werden. Dies geschieht über die Prozedur „team_j4o.setup_credentials“:

```
procedure setup_credentials(p_username in varchar2 default null,  
                           p_password in varchar2 default null,  
                           p_db_server in varchar2 default null,  
                           p_db_port in varchar2 default null,  
                           p_db_sid in varchar2 default null,  
                           p_db_service in varchar2 default null,  
                           p_db_connect_str in varchar2 default null,  
                           p_j4o_server in varchar2 default null,  
                           p_j4o_port in varchar2 default null);
```

Diese setzt *Session*-weit die angegebenen Parameter. Wird diese Prozedur mehrfach aufgerufen, so überschreiben die neueren Parameter die alten. „null“-Parameter werden ignoriert.

Es wird empfohlen, diese Prozedur am Anfang der Session aufzurufen. Insbesondere die J4O-Server- und DB-Server-spezifischen Parameter können so für alle gleich festgehalten werden.

Das folgende Beispiel eines solchen Setups zeigt, dass Benutzerinformationen getrennt von den Server-spezifischen definiert werden können:

```
team_j4o.setup_credentials(p_db_server => v_db_server,  
                           p_db_port  => v_db_port,  
                           p_db_service => v_db_service,  
                           p_j4o_server => v_j4o_server,  
                           p_j4o_port  => v_j4o_port);  
  
team_j4o.setup_credentials(p_username => v_db_username,  
                           p_password => v_db_password);
```

Sind diese Sessioneinstellungen erfolgt, kann der auszuführende Report angegeben werden:

```
procedure set_report(p_project_name in varchar2,  
                    p_report_name  in varchar2,  
                    p_format       in varchar2 default 'PDF',  
                    p_locale       in varchar2 default 'de-DE'  
);
```

Zusätzlich können Report-spezifische Parameter angelegt werden. Dies geschieht über die folgende Procedure :

```
procedure add_report_param (p_name in varchar2,  
                           p_value in varchar2);
```

Diese wird für die verschiedenen Parameter eines JasperReports jeweils einmal aufgerufen:

```
team_j4o.add_report_param ('departmentName', 'A%');  
team_j4o.add_report_param ('infoText', 'Text');
```

Um gesetzte Parameter zu löschen, kann die folgende Prozedur verwendet werden:

```
procedure clear_params;
```

Sind alle Parameter definiert (Server-, Nutzer- und Report-spezifische), kann der Report über verschiedene Funktionen abgefragt werden. Alle folgenden Funktionen haben einen Rückgabetypen „Boolean“ und mindestens die zwei Out-Parameter „p_error_code“ und „p_error_msg“.

Der Rückgabewert 0 entspricht einer erfolgreichen Durchführung und die oben genannten Out-Parameter müssen nicht ausgewertet werden.

Wenn jedoch eine 1 zurückgegeben wird, steht in „p_error_code“ der HTTP-Errorcode und in „p_error_msg“ die Fehlernachricht.

5.6.1 View URL

Der Report kann in einem beliebigen Browser über eine URL angezeigt werden. Dazu wird mit folgender Funktion eine URL generiert. Der generierte Link ist genau einmal verwendbar:

```
function get_report_view_url (p_url      out varchar2,  
                             p_err_code out number,  
                             p_err_msg  out varchar2  
    ) return boolean;
```


5.6.2 Download URL

Der Report kann mit einem beliebigen HTTP-Client über eine URL heruntergeladen werden.

Dazu wird mit folgender Funktion eine URL generiert. Der generierte Link ist genau einmal verwendbar:

```
function get_report_download_url (p_url      out varchar2,  
                                p_err_code out number,  
                                p_err_msg  out varchar2  
    ) return boolean;
```

5.6.3 Print Report

Des Weiteren, kann ein Report direkt an einen Drucker geschickt werden. Dazu kann die folgende Funktion verwendet werden:

```
function print_report (p_printer_name in varchar2,  
                      p_copies       in number default 1,  
                      p_err_code     out number,  
                      p_err_msg      out varchar2  
    ) return boolean;
```

Um die installierten Drucker zu erfahren, kann folgende Funktion ausgeführt werden:

```
function get_printers (p_printer_list out dbms_utility.lname_array)  
    return varchar2;
```

5.6.4 Mail Report

Des Weiteren, kann ein Report direkt an einen über einen konfigurierten SMTP-Server als E-Mail verschickt werden. Dazu kann die folgende Funktion verwendet werden:

```
function mail_report (  
    p_from_mail    in varchar2 default null,  
    p_to_mail      in varchar2 default null,  
    p_cc_mail      in varchar2 default null,  
    p_bcc_mail     in varchar2 default null,  
    p_subject_mail in varchar2 default null,  
    p_text_mail    in varchar2 default null,  
    p_report_name  in varchar2 default null,  
    p_err_code     out number,  
    p_err_msg      out varchar2  
) return boolean;
```

Desweiteren, existiert diese Funktion auch überladen mit „p_text_mail“ als CLOB definiert.

Die Konfiguration des SMTP-Servers kann unter 4.6 nachgeschlagen werden.

5.6.5 Lokales Speichern

Ein Report kann auch direkt auf dem Server gespeichert werden. Dazu kann folgende Funktion genutzt werden:

```
function save_report_on_server (  
    p_file_name in varchar2,  
    p_directory in varchar2 default null,  
    p_err_code  out number,  
    p_err_msg   out varchar2  
) return boolean;
```

Der Parameter „p_directory“ wird hier relativ zum Deployment-Verzeichnis interpretiert (vgl. 4.4.2)

5.6.6 Neuladen der Reportdefinitionen

Der Prozess des Neuladens aller Reports kann über die folgende Funktion angestoßen werden:

```
function redeploy return number;
```

Dabei werden alle Reports aus ihren Jar-Archiven neu geladen, um z.B. Änderungen live zu schalten.

6 Verifikation einer korrekten Installation

Um zu Verifizieren, dass der J4O-Server bzw. die PL/SQL-API korrekt installiert wurden, kann im Unterordner „Database“ die Datei „TestReportCall.sql“ gefunden werden.

In diesem werden verschiedene Wege gezeigt einen Report zu exportieren. Kopieren Sie das erste Code-Packet in einen von Ihnen gewählte PL/SQL IDE. Danach müssen die Anmeldeinformationen an Ihre Umgebung angepasst werden (siehe 5.6 „setup_credentials“).

Der ausgewählte Report steht schon mit der Standardinstallation zur Verfügung. Das ausführen des parametrisierten Scripts wird eine URL auf DBMS Output schreiben. Diese können Sie in einem beliebigen Browser öffnen.

Wenn Sie an dieser Stelle das TEAM-Logo und weitere Informationen sehen, wurde der J4O-Server korrekt installiert.