

IBM SPSS Collaboration and Deployment Services
Version 7 Release 0

Anpassungsreferenzhandbuch

IBM

Hinweis

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die Informationen unter „Bemerkungen“ auf Seite 69 gelesen werden.

Produktinformation

Diese Ausgabe bezieht sich auf Version 7, Release 0, Modifikation 0 von IBM SPSS Collaboration and Deployment Services und alle nachfolgenden Releases und Modifikationen, bis dieser Hinweis in einer Neuausgabe geändert wird.

Diese Veröffentlichung ist eine Übersetzung des Handbuchs *IBM SPSS Collaboration and Deployment Services, Version 7 Release 0, Customization Reference Guide*, herausgegeben von International Business Machines Corporation, USA

© Copyright International Business Machines Corporation 2000, 2015

Informationen, die nur für bestimmte Länder Gültigkeit haben und für Deutschland, Österreich und die Schweiz nicht zutreffen, wurden in dieser Veröffentlichung im Originaltext übernommen.

Möglicherweise sind nicht alle in dieser Übersetzung aufgeführten Produkte in Deutschland angekündigt und verfügbar; vor Entscheidungen empfiehlt sich der Kontakt mit der zuständigen IBM Geschäftsstelle.

Änderung des Textes bleibt vorbehalten.

Herausgegeben von:
TSC Germany
Kst. 2877
Januar 2015

© Copyright IBM Corporation 2000, 2015.

Inhaltsverzeichnis

Kapitel 1. Überblick über Anpassungsmöglichkeiten 1

Voraussetzungen 1

Kapitel 2. URL-Parameter 3

URL-Basispfad 3

Abfragezeichenfolge 3

 Häufige Parameter 4

 Berichtsparameter 8

 Scoring-Parameter 10

 Parameter für benutzerdefinierte Dialogfelder . . 13

HTML-Verfahren 18

Kapitel 3. IBM SPSS Collaboration and Deployment Services Tag Library 21

JSP-Architektur 22

Unterstützte Elemente 22

 Berichte 23

 Jobs 23

 Scoring-Modelle 24

 Benutzerdefinierte Dialogfelder 24

Aufbauen einer Anwendung 25

Implementierungsdetails 27

Öffentliche JavaScript-API 27

 Funktion "runRepositoryItem" 27

 Funktion "getBookmarkedValues" 28

 Funktion "retrievePromptValues" 28

IBM SPSS Collaboration and Deployment Services

Tag Library - Tagreferenz 29

 Tag "credential" 29

 Tag "repositoryItem" 32

 Tag "repositoryItemPrompt" 37

 Tag "report" 38

 Tag "reportPrompt" 38

 Tag "outputLocation" 38

 Tag "sourceLinkPrompt" 40

 Tag "sourceLinkRepositoryItem" 43

 Tag "sourceLinkReport" 44

 Tag "sourceLinkVariable" 44

 Tag "actionHandler" 44

 Tag "actionParameter" 45

Tagbibliothekbeans 46

Credential-Bean 46

ReportBean-Bean 47

SearchBean-Bean 47

PevMetaDataBean-Bean 48

ScoringBean-Bean 49

JSP-Beispiele 50

Kapitel 4. Portalintegration 51

Installation 51

Konfiguration 52

Kapitel 5. HTML-Archiv 55

Dateistruktur 55

Erstellen von HTMLC-Dateien 55

 Beispiel für eine benutzerdefinierte HTMLC-Datei 55

Kapitel 6. Anpassungsbeispiel 57

IBM SPSS Collaboration and Deployment Services

Tag Library 57

 Berichtsdefinitionen 57

 Ausführen von Berichten von BIRT Report Designer for IBM SPSS 58

 Ausführen von Visualisierungsberichten 61

 JavaScript-API 61

 Interaktivität von Visualisierungsberichten . . . 61

URL-Fragmente 62

Framework "Tab Extension" 62

Kapitel 7. Erstellen von benutzerdefinierten Datenservicetreibern 65

Datenservice-API 65

Erstellen eines benutzerdefinierten Treibers 66

Beispiel eines benutzerdefinierten Treibers 67

Bemerkungen 69

Hinweise zur Datenschutzrichtlinie 70

Marken 71

Index 73

Kapitel 1. Überblick über Anpassungsmöglichkeiten

IBM® SPSS Collaboration and Deployment Services bietet eine Vielzahl von Verfahren zum Anpassen der Interaktion mit Inhalten, die im Repository gespeichert sind. Hier einige Beispiele:

- Referenzieren von Repository-Inhalten direkt über URL-Parameter (Uniform Resource Locators).
- Erstellen von benutzerdefinierten Webseiten mithilfe von JSP-Tags (JavaServer Page) und basierend auf Informationen aus Berichten und Abfragen, die im Repository gespeichert sind.
- Einbetten von Repository-Inhalten (z. B. Berichte) in Portalseiten.
- Durchführen von Stapelverarbeitungsvorgängen für Repository-Inhalte mithilfe von Python-Skripts. Weitere Informationen finden Sie in der Dokumentation zu IBM SPSS Collaboration and Deployment Services - Essentials for Python.

Voraussetzungen

Zur korrekten Verarbeitung von benutzerdefinierten Dialogfeldern müssen die folgenden Anforderungen erfüllt sein:

- Ein ferner Ausführungsserver für IBM SPSS Statistics muss in IBM SPSS Collaboration and Deployment Services Deployment Manager eingerichtet und dann über die browserbasierte Instanz von IBM SPSS Collaboration and Deployment Services Deployment Manager als Standardserver für die Ausführung von benutzerdefinierter Dialogfeldsyntax festgelegt werden. Es ist auch möglich, individuelle benutzerdefinierte Dialogfelder zur Verwendung eines bestimmten IBM SPSS Statistics-Servers zu konfigurieren, der vom Systemstandard abweicht.
- Der Benutzer muss über die Berechtigung *Benutzerdefinierte Dialogfelder ausführen* verfügen, um benutzerdefinierte Dialogfelder ausführen zu können.
- Der Zugriff auf die Sicherungsdatei (.sav) von IBM SPSS Statistics wird vom Datendateitreiberservice von IBM SPSS Statistics aktiviert. Dieser muss mit der browserbasierten Instanz von IBM SPSS Collaboration and Deployment Services Deployment Manager installiert, gestartet und anschließend als Treiber für IBM SPSS Statistics-Daten festgelegt werden. IBM Kunden können sich diese Software herunterladen.

Wichtig: Der Datendateitreiberservice von IBM SPSS Statistics muss auf einem Host mit demselben Betriebssystemtyp wie der Repository-Host ausgeführt werden. Es ist beispielsweise nicht möglich, ein Repository auf einem Linux-Server in Verbindung mit dem Datendateitreiberservice auf einem Windows-Server zu verwenden. Informationen zur Systemkonfiguration und zu Aktionen von IBM SPSS Collaboration and Deployment Services finden Sie in der Administratordokumentation.

Kapitel 2. URL-Parameter

Sie können über direkte URLs (Uniform Resource Locators) auf Berichte von IBM SPSS Collaboration and Deployment Services Deployment Portal und andere Repository-Objekte zugreifen.

Mit URLs können Sie Berichtsdaten auf unterschiedliche Art und Weise direkt freigeben. Sie können sie z. B. in Ihre externen Websites und Anwendungen einbetten. In diesem Referenzdokument sind unterschiedliche URL-Parameter aufgeführt. Darüber hinaus enthält es einige Tipps zur Erstellung und Verwendung von URL-Abfragezeichenfolgen für IBM SPSS Collaboration and Deployment Services Deployment Portal. Hilfe erhalten Sie beim Technical Support.

Die in diesem Dokument behandelten URL-Parameter stehen in keinem Zusammenhang mit den URLs, die in IBM SPSS Collaboration and Deployment Services Deployment Manager verfügbar sind.

URL-Basispfad

Der Basispfad für alle Anforderungen lautet:

`http://<Hostname>:<Port>/<Kontextpfad>/peb/view/<Content-Repository-Pfad>`

oder

`http://<Hostname>:<Port>/<Kontextpfad>/peb/view?id=<Objekt-ID>`

<Hostname>

Name oder IP-Adresse des Repository-Servers

Anmerkung: Eine IPv6-Adresse muss in eckige Klammern eingeschlossen werden, z. B. [3ffe:2a00:100:7031::1].

<Port>

Portnummer für die Verbindung zu dem Repository-Server

<Kontextpfad>

Optionaler benutzerdefinierter Kontextpfad für den Repository-Server

<Content-Repository-Pfad>

Ressourcenpfad des Repository-Objekts, das behandelt werden soll

<Objekt-ID>

Ressourcen-ID des Repository-Objekts, das behandelt werden soll

Beispiele

`http://IhrServer:8080/peb/view/sample/employee.str`

`http://IhrServer:8080/peb/view?id=0a58c3461e885d240000010f4cc607188375`

Abfragezeichenfolge

An den Basispfad der URL-Referenzierung kann sich eine Abfragezeichenfolge mit Parametern anschließen, die zusätzliche Verarbeitungsinformationen enthalten. Die Abfragezeichenfolge beginnt mit einem Fragezeichen und enthält Parameter-/Wertepaare, die durch das Et-Zeichen (&) getrennt sind.

Beachten Sie Folgendes: Wenn ein Repository-Element von seiner Ressourcen-ID referenziert wird, ist das die Abfragezeichenfolge einleitende Fragezeichen für den Parameter `id` bereits vorhanden und sollte für andere Parameter nicht wiederholt werden.

Eine URL muss mindestens den Content-Repository-Pfad im Basispfad oder den Parameter `id` enthalten. Andere Parameter sind optional. Sofern nicht anders angegeben, werden für Parameter und ihre Werte Groß- und Kleinschreibung unterschieden. Einige Parameter wie `username` und `password` werden praktisch in allen URL-Abfragen verwendet, während die Verwendung anderer Parameter vom in der Abfrage referenzierten Elementtyp abhängig sein kann. Beachten Sie, dass das System so konfiguriert sein kann, dass es einen benutzerspezifischen Authentifizierungsmechanismus verwendet, damit in der Abfragezeichenfolge keine sicheren Anmeldeparameter angegeben werden müssen.

Reservierte Zeichen wie `&` und ausgeschlossene US-ASCII-Zeichen wie `#` sollten in der URL codiert werden, bevor sie als Parameterwert in der Abfragezeichenfolge verwendet werden. Zeichen im reservierten Satz sind jedoch nicht in allen Kontexten reserviert. Im Allgemeinen ist ein Zeichen reserviert, wenn sich die Semantik des URI ändert, wenn das Zeichen mit seiner vorangestellten US-ASCII-Codierung ersetzt wird. Daher sind bestimmte Zeichen wie `?`, `=` und `:` in den Parameterwerten nicht reserviert, während Zeichen wie `&` und `#` reserviert sind und daher in der URL codiert werden müssen.

Das Zeichen `&` sollte beispielsweise in der URL als `%26` codiert werden. Beispiel für die Angabe der folgenden URL:

```
http://IhrServer:8080/peb/view/sample/employee.str?username=testuser&admin
```

sollte angegeben werden als

```
http://IhrServer:8080/peb/view/sample/employee.str?username=testuser%26admin
```

In den folgenden Abschnitten werden die Parameter einzeln beschrieben.

Häufige Parameter

Häufige Parameter werden in praktisch allen URL-Referenzierungen oder für mehrere Typen von Repository-Elementen verwendet.

Parameter `id`

Der Parameter `id` legt die Repository-ID für das Element fest, das behandelt werden soll.

Syntax

```
id=<ID>
```

Der Wert `<ID>` entspricht der ID des Repository-Objekts.

Beispiel

```
http://IhrServer:8080/peb/view?id=0a58c3461e885d240000010f4cc607188375
```

Parameter `version`

Der Parameter `version` legt die Version des Repository-Objekts fest, das unter Verwendung der Versionsmarkierung bzw. der Versionsbeschriftung behandelt werden soll.

Sonderzeichen (z. B. Leerzeichen) muss ein umgekehrter Schrägstrich vorangestellt werden. Lassen Sie diesen Parameter weg, um die AKTUELLSTE Version anzuzeigen.

Syntax

```
version=m.<Versionsmarkierung>  
version=l.<Beschriftung>
```

Der Wert `<Versionsmarkierung>` entspricht der Version des Repository-Objekts. Alternativ kennzeichnet der Wert `<Beschriftung>` die Versionsbeschriftung des Repository-Objekts.

Beispiele

`http://srvr:80/peb/view/sample/emp.str?version=m.1:2006-12-04%2020:39:17.995`

`http://IhrServer:8080/peb/view/sample/employee.str?version=1.firstVersion`

Parameter username

Der Parameter `username` definiert den Benutzer, mit dem die Anmeldung beim System erfolgt.

Syntax

`username=<Benutzer-ID>`

`username=<Benutzer-ID>`

Der Wert `<Benutzer-ID>` ist die Benutzer-ID der Person, die sich beim Repository-Server anmeldet.

Beispiel

`http://IhrServer:8080/peb/view/sample/employee.str?username=validUser
&password=pass&provider=Native`

Parameter password

Der Parameter `password` definiert das Kennwort, mit dem die Anmeldung beim System erfolgt.

Syntax

`password=<Kennwort>`

Der Wert `<Kennwort>` ist das Kennwort der Person, die sich beim Repository-Server anmeldet.

Beispiel

`http://IhrServer:8080/peb/view/sample/employee.str?username=validUser
&password=pass&provider=Native`

Parameter provider

Der Parameter `provider` definiert den Sicherheitsprovider, mit dem die Berechtigungsnachweise validiert werden.

Für `provider` muss ein Wert angegeben werden, falls die Parameter `username` und `password` verwendet werden.

Syntax

`provider=<Provider>`

Der Wert `<Provider>` gibt den Sicherheitsprovider an. Gültige Werte sind:

- `Native` für den integrierten Anbieter.
- `AD_<Name>/<Domäne>` für Active Directory, wobei `<Name>` dem Namen des Sicherheitsproviders im System und `<Domäne>` dem DNS-Namespaces entspricht

Sonderzeichen (z. B. Leerzeichen) muss ein umgekehrter Schrägstrich vorangestellt werden.

Beispiel

`http://IhrServer:8080/peb/view/sample/employee.str
?username=validUser&password=pass&provider=Native`

Parameter promptstate

Der Parameter `promptstate` legt fest, ob das Dialogfeld, das zur Eingabe von Variablenwerten auffordert, die nicht in der Abfragezeichenfolge angegeben werden, unterdrückt werden soll.

Syntax

promptstate=<x>

Der Wert 1 für <x> unterdrückt das Dialogfeld mit der Eingabeaufforderung während der Laufzeit und verwendet den angegebenen Variablenwert für angeforderte Variablen, die nicht angegeben wurden. Beim Wert 2 wird das Dialogfeld mit der Eingabeaufforderung für alle angeforderten Variablen angezeigt, die nicht angegeben wurden. Alternativ können Sie diesen Parameter weglassen, um die Anzeige des Dialogfelds mit der Eingabeaufforderung zuzulassen.

Beispiel

```
http://IhrServer:8080/peb/view/sample/employee.viz?username=validUser
&password=pass&provider=Native&fragment=true&outputtype=html
&var_EmployeeID=1&promptstate=1
```

Parameter waitstate

Der Parameter waitstate legt fest, ob das Fenster "Warten" angezeigt wird, wenn ein Bericht ausgeführt wird.

Syntax

waitstate=<x>

Beim Wert 1 für <x> wird das Fenster "Warten" unterdrückt. Lassen Sie diesen Parameter weg, um das Fenster "Warten" anzuzeigen.

Beispiel

```
http://IhrServer:8080/peb/view/sample/employee.viz?username=validUser
&password=pass&provider=Native&fragment=true&outputtype=html
&var_EmployeeID=1&promptstate=1&waitstate=1&fragment=true
```

Parameter partId

Der Parameter partId gibt einen bestimmten Teil des referenzierten Repository-Objekts an.

Bei HTMLC-Dateien kann dieser Parameter eine bestimmte Datei innerhalb des Archivs referenzieren. Bei IBM SPSS Statistics-Ausgabedateien (.spw) entspricht der Parameter dem Index gemäß der Anzeige in der Gliederung der Datei. Geben Sie also partId=0 an, um den ersten Teil abzurufen.

Syntax

partId=<Referenz-ID>

<Referenz-ID> kann folgende Werte annehmen:

- Den relativen Pfad und den Namen einer Datei innerhalb einer HTMLC-Datei.
- Den Index der gewünschten Ausgabe in einer .spw-Datei

Beispiel

```
http://IhrServer:8080/peb/view/output.htmlc?username=validUser
&password=pass&provider=Native&partId=img/chart.png
```

```
http://IhrServer:8080/peb/view/output.spw?username=validUser
&password=pass&provider=Native&partId=1
```

Parameter outputtype

Der Parameter outputtype legt den Dateityp des Ergebnissets fest.

Syntax

outputtype=<Dateityp>

Dabei entspricht <Dateityp> einem der Werte in den folgender Tabellen.

Tabelle 1. Ausgabetypen von BIRT Report Designer for IBM SPSS.

Wert	Rückgabe
HTML	HTML
Excel 97/2003	Microsoft Excel
PowerPoint	Microsoft PowerPoint
Word-Dokument	Microsoft Word
PDF	Portable Document Format
PDF - An Seitenbreite anpassen	Portable Document Format mit Breitenvergrößerung
PDF – Nur Seitennummerierung bei Seitenumbruch	Portable Document Format mit Seitennummerierung bei Seitenumbruch
PDF – In ganze Seite einpassen	Portable Document Format mit Seitenvergrößerung
PostScript	PostScript-Format
PostScript – An Seitenbreite anpassen	PostScript-Format mit Breitenvergrößerung
PostScript - Nur Seitennummerierung bei Seitenumbruch	PostScript-Format mit Seitennummerierung bei Seitenumbruch
PostScript – In ganze Seite einpassen	PostScript-Format mit Seitenvergrößerung
BIRT-RPT-Dokument	Berichtdokument von BIRT Report Designer for IBM SPSS
ask	Eine Eingabeaufforderung, die den Benutzer während der Laufzeit zur Angabe eines Ausgabeformats auffordert

Tabelle 2. Visualisierungsausgabetypen.

Wert	Rückgabe
png	Portable Network Graphics-Format
emf	Enhanced Metafile-Format
jpeg	JPEG
html	HTML. Dies ist für Visualisierungsberichte nur ein gültiges Ausgabeformat, wenn es sich bei der Ausgabe um eine Tabelle handelt. Wenn HTML als Format für einen Visualisierungsbericht verwendet wird, der keine Tabelle erzeugt, wird die Ausgabe in ein PNG-Bild konvertiert.
pdf	PDF
ask	Eine Eingabeaufforderung, die den Benutzer während der Laufzeit zur Angabe eines Ausgabeformats auffordert

Tabelle 3. Ausgabetypen des benutzerdefinierten Dialogs.

Wert	Rückgabe
SPW	IBM SPSS Statistics-Web-Ausgabe-Viewer
HTML	HTML

Beispiel

<http://IhrServer:8080/peb/view/sample/employee.viz?username=validUser&password=pass&provider=Native&fragment=true&outputtype=html>

Parameter format

Der Parameter format legt fest, ob die ursprüngliche, im Repository gespeicherte Datei zurückgegeben oder ausgeführt werden soll.

Syntax

format=raw

Bei Angabe des Werts *raw* wird die ursprüngliche Datei zurückgegeben. Beispielsweise wird bei Berichten von BIRT Report Designer for IBM SPSS über den Parameter *format* die ursprüngliche *.rptdesign-Datei zurückgegeben und die Abfrage nicht dynamisch ausgeführt.

Beispiel

```
http://IhrServer:8080/peb/view/sample/employee.rptdesign?username=validUser
&password=pass&provider=Native&format=raw
```

Parameter fragment

Der Parameter *fragment* legt fest, ob die Benutzerschnittstellenelemente von IBM SPSS Collaboration and Deployment Services Deployment Portal (also die Kopf- und Fußzeile und der Content-Repository-Baum) mit den Berichtsergebnissen angezeigt werden sollen.

Syntax

fragment=true

Der Wert *true* unterdrückt die Elemente der IBM SPSS Collaboration and Deployment Services Deployment Portal-Schnittstelle. Lassen Sie diesen Parameter weg, um die Schnittstelle anzuzeigen.

Beispiel

```
http://IhrServer:8080/peb/view/sample/employee.viz?username=validUser
&password=pass&provider=Native&fragment=true
```

Parameter für Variablen

Bei nicht für einen Bericht bestimmten Repository-Elementen, die Variablen verwenden (z. B. Jobs), kann der Wert einer Variablen angegeben werden, indem Name und Wert der Variablen in der Abfragezeichenfolge der URL mit angegeben werden.

Bei Berichtselementen muss dem Variablennamen das Präfix *var_* vorangestellt werden.

Syntax

<Variable>=<Wert>

Der Wert <Variable> entspricht dem Namen der Variablen, die erfüllt werden muss. <Wert> ist der Eintrag, der verwendet werden muss, um die angegebene Berichtsvariable zu erfüllen.

Beispiel

```
http://IhrServer:8080/peb/view/sample/myJob?username=validUser
&password=pass&provider=Native&region=1
```

Berichtsparameter

Berichtsparameter werden in Referenzen auf in IBM SPSS Collaboration and Deployment Services Repository gespeicherte Berichte verwendet.

Bei den Berichten kann es sich um Visualisierungsberichte oder Berichte von BIRT Report Designer for IBM SPSS handeln.

Parameter dbcredential_datasourcename

Der Parameter *dbcredential_datasourcename* legt die Berechtigungsnachweise fest, mit denen die Anmeldung in der Datenquelle erfolgt.

Dieser Parameter wird verwendet, wenn die Benutzer-ID für die Datenquelle nicht mit der Benutzer-ID von IBM SPSS Collaboration and Deployment Services Deployment Portal identisch ist.

Syntax

`dbcredential_<Datenquellename>=<Anmelde-ID>`

Der Wert `<Datenquellename>` ist der Name der angegebenen Datenquelle. Der Wert `<Anmelde-ID>` ist die ID des Berechtigungsnachweisobjekts, das für die Verbindung zu der Datenquelle verwendet wird.

Beispiel

```
http://IhrServer:8080/peb/view/sample/employee.viz
?dbcredential_yourDS=0a58c346cd5b72010000010f3df6d5e28130
```

Parameter `dbuser_datasourcename`

Der Parameter `dbuser_datasourcename` legt die Benutzer-ID fest, mit der die Anmeldung in der Datenquelle erfolgt.

Er wird verwendet, wenn die Benutzer-ID für die Datenquelle von der Benutzer-ID für IBM SPSS Collaboration and Deployment Services Deployment Portal abweicht.

Syntax

`dbuser_<Datenquellename>=<Benutzer-ID>`

Der Wert `<Datenquellename>` steht für den Namen der angegebenen Datenquelle. Der Wert `<Benutzer-ID>` steht für die Benutzer-ID der Person, die sich bei der Datenquelle anmeldet.

Beispiel

```
http://IhrServer:8080/peb/view/sample/employee.viz?dbuser_yourDS=sa
```

Parameter `dbpwd_datasourcename`

Der Parameter `dbpwd_datasourcename` legt das Kennwort fest, mit dem die Anmeldung in der Datenquelle erfolgt.

Dieser Parameter wird verwendet, wenn die Benutzer-ID für die Datenquelle nicht mit der Benutzer-ID von IBM SPSS Collaboration and Deployment Services Deployment Portal identisch ist.

Syntax

`dbpwd_<Datenquellename>=<Kennwort>`

Der Wert `<Datenquellename>` ist der Name der angegebenen Datenquelle. Der Wert `<Kennwort>` ist das Kennwort der Person, die sich bei der Datenquelle anmeldet.

Beispiel

```
http://IhrServer:8080/peb/view/sample/employee.viz
?dbuser_yourDB=sa&dbpwd_yourDB=sa
```

Anmerkung: Wenn der Parameter `dbcredential_datasourcename` angegeben wurde, wird dieser Parameter vor den Parametern `dbuser_datasourcename` und `dbpwd_datasourcename` für die Herstellung einer Verbindung zu der Datenquelle berücksichtigt.

Parameter `width`

Der Parameter `width` legt die Breite eines Bilds oder Diagramms fest. Dieser Parameter wird vor allem für Visualisierungsberichte verwendet.

Für Berichte mit Höhen- und Breitenangaben müssen die Parameter `height` und `width` angegeben werden. Wenn einer der Parameter fehlt, wird das Diagramm mit der Standardhöhe und -breite angezeigt.

Syntax

`width=<x>`

Der Wert `<x>` gibt den ganzzahligen Wert für die Breite in Pixeln an.

Beispiel

```
http://IhrServer:8080/peb/view/sample/employee.viz?username=validUser
&password=pass&provider=Native&fragment=true&outputtype=html
&var_EmployeeID=1&promptstate=1&waitstate=1&width=500&height=1000
```

Parameter height

Der Parameter `height` legt die Höhe eines Bilds oder Diagramms fest. Dieser Parameter wird vor allem für Visualisierungsberichte verwendet.

Für Berichte mit Höhen- und Breitenangaben müssen die Parameter `height` und `width` angegeben werden. Wenn einer der Parameter fehlt, wird das Diagramm mit der Standardhöhe und -breite angezeigt.

Syntax

`height=<x>`

Der Wert `<x>` gibt den ganzzahligen Wert für die Höhe in Pixeln an.

Beispiel

```
http://IhrServer:8080/peb/view/sample/employee.viz?username=validUser
&password=pass&provider=Native&fragment=true&outputtype=html
&var_EmployeeID=1&PROMPTSTATE=1&waitstate=1&width=500&height=1000
```

Parameter var_variable

Der Parameter `var_variable` legt den Wert fest, der die angegebene Berichtsvariable erfüllt.

Syntax

`var_<Variable>=<Wert>`

Der Wert `<Variable>` ist der Name der Variablen, die an den Bericht übergeben wird.

Beispiel

```
http://IhrServer:8080/peb/view/sample/employee.rptdesign?username=validUser
&password=pass&provider=Native&fragment=true&outputtype=html&var_EmployeeID=1
```

Hinweise

- Bei Berichten unterdrückt die Angabe eines Variablenwerts in der URL das Eingabeaufforderungsdialogfeld für diese Variable.
- Um einen einzelnen Variablenwert anzugeben (=), verwenden Sie die Syntax `var_Lastname=Curtis`.
- Um mehrere Variablenwerte anzugeben (IN), verwenden Sie die Syntax `var_Lastname=Curtis &var_Lastname=McLind`.
- Um einen Bereich von Variablenwerten anzugeben (BETWEEN), verwenden Sie die Syntax `var_Dateship=3-1-2007&var_Dateship=3-31-2007`.
- Um Werte für mehrere Variablen anzugeben, verwenden Sie die Syntax `var_Lastname=Curtis &var_Dateship=3-1-2007&var_Dateship=3-31-2007`.

Scoring-Parameter

Scoring-Parameter werden bei der Referenzierung von Scoring-Konfigurationen zum Generieren von Scores verwendet.

Parameter dataset

Der Parameter `dataset` gibt den Speicherort einer SQL-Datenproviderdefinition an, die für das Stapelscoring verwendet werden soll.

Der Wert dieses Parameters ist ein relativer Pfad in IBM SPSS Collaboration and Deployment Services Repository.

Syntax

```
dataset=<DPD-Speicherort>  
dataset.<Tabellen-ID>=<DPD-Speicherort>
```

Der Wert `<DPD-Speicherort>` steht für den Repository-Pfad zur Datenproviderdefinition.

Wenn das Dataset mehrere Tabellen enthält, hängen Sie an den Parameter einen Punkt gefolgt von der Tabellen-ID für die gewünschte Tabelle an. Der Wert `<Tabellen-ID>` kann durch Untersuchung des Scoring-Modells in IBM SPSS Modeler ermittelt werden.

Beispiel

```
http://IhrServer:8080/peb/view/myPMML.xml?username=validUser  
&password=pass&scoring_configuration=testConfig  
&dataset=/datasets/dataset.sqldpd
```

Parameter dataset_label

Mit dem Parameter `dataset_label` kann der Benutzer die entsprechende Version des Datasets angeben. Die angegebene Datasetversion muss mit dem in der Scoring-Konfiguration definierten Datenprovider kompatibel sein.

Ohne Angabe wird die *AKTUELLSTE* Version verwendet.

Syntax

```
dataset_label=<meineBeschriftung>  
dataset_label.<Tabellen-ID>=<meineBeschriftung>
```

Der Wert `<meineBeschriftung>` steht für die Beschriftung für die Datasetversion.

Wenn das Dataset mehrere Tabellen enthält, hängen Sie an den Parameter einen Punkt gefolgt von der Tabellen-ID für die gewünschte Tabelle an. Der Wert `<Tabellen-ID>` kann durch Untersuchung des Scoring-Modells in IBM SPSS Modeler ermittelt werden.

Beispiel

```
http://IhrServer:8080/peb/view/myPMML.xml?username=validUser  
&password=pass&scoring_configuration=testConfig  
&dataset=/datasets/dataset.sqldpd&dataset_label=PRODUCTION
```

Parameter dataset_table

Mit dem Parameter `dataset_table` kann der Benutzer eine Tabelle innerhalb eines Datasets angeben.

Syntax

```
dataset_table=<meineTabelle>  
dataset_table.<Tabellen-ID>=<meineTabelle>
```

Der Wert `<meineTabelle>` steht für den Namen der Datasettabelle.

Wenn das Dataset mehrere Tabellen enthält, hängen Sie an den Parameter einen Punkt gefolgt von der Tabellen-ID für die gewünschte Tabelle an. Der Wert `<Tabellen-ID>` kann durch Untersuchung des Scoring-Modells in IBM SPSS Modeler ermittelt werden.

Beispiel

```
http://IhrServer:8080/peb/view/myPMML.xml?username=validUser
&password=pass&scoring_configuration=testConfig&fragment=true&
dataset=/data/mySet&
dataset_label=PRODUCTION&
dataset_rowlimit=2&
dataset_table=myTable&
promptstate=1
```

Parameter `dataset_rowlimit`

Der Benutzer kann die Menge der Daten, die vom Dataset für das Stapelscoring verarbeitet werden, begrenzen. Auf diese Weise lassen sich Prozesse mit langen Laufzeiten vermeiden.

Der Parameter `dataset_rowlimit` gibt die Anzahl der Datenzeilen an, die aus dem Dataset extrahiert werden.

Syntax

```
dataset_rowlimit=<x>
dataset_rowlimit.<Tabellen-ID>=<x>
```

Der Wert `<x>` steht für die Anzahl der Datasetzeilen, die extrahiert werden sollen.

Wenn das Dataset mehrere Tabellen enthält, hängen Sie an den Parameter einen Punkt gefolgt von der Tabellen-ID für die gewünschte Tabelle an. Der Wert `<Tabellen-ID>` kann durch Untersuchung des Scoring-Modells in IBM SPSS Modeler ermittelt werden.

Beispiel

```
http://IhrServer:8080/peb/view/myPMML.xml?username=validUser
&password=pass&scoring_configuration=testConfig
&dataset=/datasets/dataset.sqldpd&dataset_rowlimit=1000
```

Parameter `scoring_configuration`

Der Parameter `scoring_configuration` gibt die Scoring-Konfiguration an, die von der Scoring-Engine für das Scoring des angegebenen Modells verwendet wird.

Syntax

```
scoring_configuration=<Konfigurationsname>
```

Der Wert `<Konfigurationsname>` steht für den Namen der für das Scoring zu verwendenden Scoring-Konfiguration. Die angegebene Konfiguration muss in der Lage sein, eine Scoring-Anforderung zu verarbeiten. Ein Verweis auf eine ausgesetzte Konfiguration kann keine Scores erzeugen.

Beispiel

```
http://IhrServer:8080/peb/view/myPMML.xml?username=validUser
&password=pass&scoring_configuration=testConfig
&dataset=/datasets/dataset.sqldpd
```

Parameter `batch_type`

Der Parameter `batch_type` gibt an, welche Scoring-Eingabeaufforderungen angezeigt werden sollen.

Wenn der Parameter `dataset` angibt, generiert die Scoring-Schnittstelle die Eingabeaufforderungen für das Dataset und die Beschriftung. Falls der Parameter `batch_type` nicht angegeben wird und keine Parameter-eingaben definiert werden, wird die auf den Scoring-Parametern basierende Schnittstelle verwendet.

Syntax

```
batch_type=<Eingabeaufforderung>
```

Der Wert `<Eingabeaufforderung>` gibt die Quelle für die Eingabeaufforderungen an. Derzeit wird nur `dataset` als Quelle unterstützt. Lassen Sie diesen Parameter weg, um den Benutzer zur Eingabe von auf Parametern basierenden Werten aufzufordern.

Beispiel

```
http://IhrServer:8080/peb/view/myPMML.xml?username=validUser
&password=pass&scoring_configuration=testConfig&batch_type=dataset
```

Parameter für benutzerdefinierte Dialogfelder

Parameter für benutzerdefinierte Dialogfelder werden zur Referenzierung von Dateien für benutzerdefinierte Dialogfelder (.spd) verwendet.

Für diese Funktionalität sind IBM SPSS Statistics-Adapter in der Umgebung von IBM SPSS Collaboration and Deployment Services erforderlich. Weitere Informationen finden Sie in der Installationsdokumentation zu IBM SPSS Statistics.

Parameter `dataset.uri`

Der URI des Datasets, das vom benutzerdefinierten Dialogfeld verwendet werden soll. Bei DPDs und SAV-Dateien in IBM SPSS Collaboration and Deployment Services Repository kann der URI als Repository-Pfad oder als Ressourcen-ID angegeben werden. Wenn der URI eine Datei im Dateisystem referenziert, muss der Pfad zur Datei ein Pfad sein, der für den Server mit dem Datendateitreiber IBM SPSS Statistics, der zum Abrufen der Variablenmetadaten verwendet wird, gültig ist. Außerdem muss es ein gültiger Pfad auf dem IBM SPSS Statistics-Server sein, der die Syntax ausführen wird. Wenn ein Datasetobjekt eines Repositorys verwendet wird, kann die Version des Objekts entweder als Versionsmarkierung oder als Beschriftung an den URI angehängt werden.

Syntax

```
dataset.uri=<meinURI>
```

Der Wert <meinURI> steht für den URI für das Dataset.

Beispiel

```
http://IhrServer:8080/peb/view/myDialog.spd
?dataset.uri=spsscr:///Datasets/SpecificURI.sav
```

```
http://IhrServer:8080/peb/view/myDialog.spd
?dataset.uri=spsscr:///?id=0a30063bc975ede40000011cafb8deda8327.
```

```
http://IhrServer:8080/peb/view/myDialog.spd
?dataset.uri=file:///C:/Program%20Files/SPSSInc/Samples/accidents.sav
```

Parameter `dataset.table`

Für Datenquellen von IBM SPSS Collaboration and Deployment Services Enterprise View die Tabelle, die vom benutzerdefinierten Dialogfeld verwendet werden soll. Wenn kein Name angegeben wird, wird der Benutzer aufgefordert, eine Tabelle aus der Liste der in der DPD verfügbaren Tabellen auszuwählen.

Syntax

```
dataset.table=<meineTabelle>
```

Der Wert <meineTabelle> gibt die zu verwendende Tabelle an.

Beispiel

```
http://IhrServer:8080/peb/view/myDialog.spd
?dataset.uri=spsscr:///DPDs/myDPD&dataset.table=myTableName
```

Parameter `dataset.prompt`

Gibt an, dass der Benutzer gezwungen wird, ein Dataset für die benutzerdefinierten Dialogfelder auszuwählen. Andernfalls wird das Dataset, das der Benutzer während der Sitzung für das erste Dialogfeld geöffnet hat und das den Suchkriterien entspricht, für jedes nachfolgende benutzerdefinierte Dialogfeld verwendet, das nicht für die Verwendung eines speziellen Datasets konfiguriert wurde.

Syntax

```
dataset.prompt=<Indikator>
```

Der Wert <Indikator> ist entweder *true* oder *false*.

Beispiel

```
http://IhrServer:8080/peb/view/myDialog.spd?dataset.prompt=true
```

Parameter `dataset.search.criteria`

Die zu verwendenden Suchkriterien für die Erstellung einer Liste von Datasets während der Ausführungszeit. Die gesamte Suchzeichenfolge muss in einer einzigen Zeile eingegeben werden. Mehrere Bedingungen können mithilfe von Klammern, logischen Und-Zeichen und logischen Oder-Zeichen kombiniert werden.

Suchkriterien

```
$$repository/title_field_name=<Objektname>
```

```
$$search/mimetype=<Objekt-MIME-Typ>
```

```
$$repository/version_created_by_field=<Stempel des Benutzers, der das Objekt erstellt hat>
```

```
$$repository/version_created_date_field=<Datum der Versionserstellung>
```

```
$$repository/description_field_name=<Objektbeschreibung>
```

```
$$repository/object_last_modified_by=<Stempel des Benutzers, der das Objekt erstellt hat>
```

Syntax

```
dataset.search.criteria=<meineKriterien>
```

Der Wert <meineKriterien> entspricht dem Suchausdruck.

Beispiel

```
# sucht alle DPDS
http://IhrServer:8080/peb/view/myDialog.spd
?dataset.search.criteria='$$search/mimetype%3Dapplication/x-vnd.spss-data-provider'

# lokalisiert alle SAV-Dateien
http://IhrServer:8080/peb/view/myDialog.spd
?dataset.search.criteria=
'$$search/mimetype%3Dapplication/x-vnd.spss-spss-data%20or%20
$$search/mimetype%3Dapplication/x-vnd.spss-statistics-data'

# sucht alle Dateien, die dem Schlüsselwort SPECIAL_DATASET entsprechen
http://IhrServer:8080/peb/view/myDialog.spd
?dataset.search.criteria='$$repository/keyword_field_name%3D%3DSPECIAL_DATASET'
```

Parameter `"variable.display"`

Der Parameter `variable.display` gibt an, ob Variablennamen bzw. -beschriftungen angezeigt werden sollen oder nicht.

Syntax

```
variable.display=<Typ>
```

Der Wert <Typ> ist entweder *names*, womit Variablennamen angezeigt werden, oder *labels*, womit Variablenbeschriftungen angezeigt werden.

Beispiel

```
http://IhrServer:8080/peb/view/myDialog.spd  
?dataset.uri=spsscr:///Datasets/SpecificURI.sav&variable.display=labels
```

Parameter `variable.sort`

Der Parameter `variable.sort` gibt das zur Sortierung der Variablen verwendete Suchkriterium an.

Syntax

```
variable.sort=<meineKriterien>
```

<meineKriterien> kann folgende Werte annehmen:

- *none*, um keine zusätzliche Sortierung der ursprünglichen Reihenfolge der Daten durchzuführen
- *alphanumeric* für eine alphanumerische Sortierung von Feldnamen oder -beschriftungen (je nach Anzeige)
- *measurement* für eine Sortierung nach dem Messniveau des Felds

Beispiel

```
http://IhrServer:8080/peb/view/myDialog.spd  
?dataset.uri=spsscr:///Datasets/SpecificURI.sav&variable.sort=alphanumeric
```

Parameter `stylesheet.url`

Wenn Sie ein CSS-Style-Sheet verwenden, das im Repository gespeichert ist, die Repository-URL des Style-Sheets.

Syntax

```
stylesheet.url=<meineURL>
```

Der Wert <meineURL> ist die URL für das Style-Sheet.

Beispiel

```
http://IhrServer:8080/peb/view/myDialog.spd  
?stylesheet.url=/peb/view/EditBox_pes.css&fragment=true
```

Parameter `stylesheet.name`

Wenn Sie ein CSS-Style-Sheet verwenden, das in die Datei des benutzerdefinierten Dialogfelds eingebettet ist, der Name des Style-Sheets. Das Style-Sheet kann der benutzerdefinierten Dialogdatei hinzugefügt werden, indem Komprimierprogramme wie WinZip verwendet werden.

Syntax

```
stylesheet.name=<meineStile>
```

Der Wert <meineStile> gibt den Namen des Style-Sheets an.

Beispiel

```
http://IhrServer:8080/peb/view/myDialog.spd  
?stylesheet.name=EditBox.css
```

Parameter `javascript.url`

Wenn Sie ein JavaScript verwenden, das im Repository gespeichert ist, die Repository-URL der Scriptdatei.

Syntax

```
javascript.url=<meineURL>
```

Der Wert <meineURL> steht für die URL für die JavaScript-Datei.

Beispiel

```
http://IhrServer:8080/peb/view/myDialog.spd
?javascript.url=/peb/view/EditBox_pes.js&fragment=true
```

Parameter javascript.name

Wenn Sie ein JavaScript-Sheet verwenden, das in die Datei des benutzerdefinierten Dialogfelds eingebettet ist, der Name der Scriptdatei.

Syntax

```
javascript.name=<meineDatei>
```

Der Wert <meineDatei> ist der Name der JavaScript-Datei.

Beispiel

```
http://IhrServer:8080/peb/view/myDialog.spd?javascript.name=EditBox.js
```

Parameter validate.method

Ein Validierungsverfahren aus der angegebenen JavaScript-Datei, das aufgerufen wird, bevor die Seite angezeigt wird. Das übergebene Formular sollte der einzige Parameter für die Methode sein. Nach Auswertung der Eingaben gibt das Verfahren einen booleschen Wert zurück. Die Methode sollte "true" zurückgeben, falls alles korrekt ist, und "false", falls die Übertragung abgebrochen werden soll.

Syntax

```
validate.method=<meineMethode>
```

Der Wert <meineMethode> ist der Name der für die Validierung zu verwendenden Methode in der JavaScript-Datei.

Beispiel

```
http://IhrServer:8080/peb/view/myDialog.spd?javascript.name=EditBox.js
&validate.method=myValidate
```

Parameter output.format

Das Format der zu erzeugenden Ausgabe. Standardformat ist das Format des Web-Ausgabe-Viewers von IBM SPSS Statistics (*.spw*). In einigen Fällen kann es sinnvoll sein, stattdessen HTML zu erzeugen. Beim Ausgabeformat wird zwischen Groß- und Kleinschreibung unterschieden.

Dieser Parameter gibt dieselben Informationen wie der Parameter `outputtype` an, wird jedoch nur für benutzerdefinierte Dialogfelder berücksichtigt.

Syntax

```
output.format=<meinFormat>
```

Der Wert <meinFormat> ist das Format für die Ausgabe. Folgende Werte sind gültig:

- *SPW* für den Web-Ausgabe-Viewer IBM SPSS Statistics
- *HTML* für eine HTML-Ausgabe

Beispiel

```
http://IhrServer:8080/peb/view/myDialog.spd?output.format=SPW
```

Parameter output.filename

Der Name der Ausgabedatei. Wenn nicht angegeben, erhält die generierte Ausgabedatei den Namen der Datei des benutzerdefinierten Dialogfelds, jedoch ohne die Erweiterung *.spw*.

Syntax

`output.filename=<meineDatei>`

Der Wert `<meineDatei>` ist der Name für die Ausgabedatei.

Beispiel

`http://IhrServer:8080/peb/view/myDialog.spd?output.filename=MyOutputName.spw`

Parameter `showOutline`

Gibt an, ob die Gliederung angezeigt werden soll. Der Standardwert lautet `true`.

Syntax

`showOutline=<Indikator>`

Der Wert von `<Indikator>` ist entweder `true` oder `false`.

Beispiel

`http://IhrServer:8080/peb/view/myDialog.spd?showOutline=true`

Parameter `allowPivoting`

Gibt an, ob es möglich sein soll, die Tabelle zu bearbeiten. Wenn die Option inaktiviert ist, kann der Benutzer Schichten nicht drehen, vertauschen oder ändern, Ansichten nicht speichern und Daten nicht in einem neuen Fenster öffnen. Der Standardwert lautet `true`.

Syntax

`allowPivoting=<Indikator>`

Der Wert von `<Indikator>` ist entweder `true` oder `false`.

Beispiel

`http://IhrServer:8080/peb/view/myDialog.spd?allowPivoting=true`

Parameter `allowPrinterFriendly`

Gibt an, ob eine druckerfreundliche Version für eine bestimmte Tabelle geöffnet werden kann. Der Standardwert lautet `true`.

Syntax

`allowPrinterFriendly=<Indikator>`

Der Wert von `<Indikator>` ist entweder `true` oder `false`.

Beispiel

`http://IhrServer:8080/peb/view/myDialog.spd?allowPrinterFriendly=true`

Parameter `allowDownload`

Gibt an, ob die Daten in eine lokale Datei heruntergeladen werden können. Der Standardwert lautet `true`.

Syntax

`allowDownload=<Indikator>`

Der Wert von `<Indikator>` ist entweder `true` oder `false`.

Beispiel

`http://IhrServer:8080/peb/view/myDialog.spd?allowDownload=true`

Parameter showLogs

Gibt an, ob Protokolleinträge in der Ausgabe angezeigt werden sollen. Der Standardwert lautet `true`.

Syntax

```
showLogs=<Indikator>
```

Der Wert von `<Indikator>` ist entweder `true` oder `false`.

Beispiel

```
http://IhrServer:8080/peb/view/myDialog.spd?showLogs=true
```

Parameter statistics.server

IBM SPSS Statistics-Server, der zur Ausführung der Syntax des benutzerdefinierten Dialogfelds verwendet wird. Der Wert kann ein URI oder ein Name sein, der auf einen in IBM SPSS Collaboration and Deployment Services definierten Server verweist. Bei mehreren Servern kann dieser Wert der URI oder Name eines Server-Clusters sein.

Syntax

```
statistics.server=<Server-ID>
```

Der Wert `<Server-ID>` gibt den für die Ausführung zu verwendenden Server an.

Beispiel

```
http://IhrServer:8080/peb/view/myDialog.spd?  
statistics.server=spsscr:///id=0a30063bc975ede40000011cafb8deda8327
```

```
http://IhrServer:8080/peb/view/myDialog.spd  
?statistics.server=localStatisticsServer
```

```
http://IhrServer:8080/peb/view/myDialog.spd  
?statistics.server=copServerCluster
```

Parameter statistics.server.credential

Die Berechtigungsnachweise, die verwendet werden sollen, um beim Ausführen der Syntax eine Verbindung zum IBM SPSS Statistics-Server aufzubauen. Der Wert kann ein URI oder ein Name sein, der auf eine Berechtigungsnachweise für IBM SPSS Collaboration and Deployment Services verweist.

Syntax

```
statistics.server.credential=<meineBerechtigungsnachweise>
```

Der Wert `<meineBerechtigungsnachweise>` gibt die Berechtigungsnachweise an, unter denen die Ausführung stattfindet.

Beispiel

```
http://IhrServer:8080/peb/view/myDialog.spd?statistics.server=localStatisticsServer  
&statistics.server.credential=spsscr:///id=0a30063bc975ede40000011cafb8deda8327.
```

```
http://IhrServer:8080/peb/view/myDialog.spd?statistics.server=localStatisticsServer  
&statistics.server.credential=administrator
```

HTML-Verfahren

Verwenden eines HTML-Editors

Viele HTML-Editoren können die Erstellung von URL-Abfragezeichenfolgen vereinfachen und die korrekten Trennzeichen zwischen Parameter setzen.

Verwenden von HTML-Formularen zum Übergeben von Anforderungen

Anforderungen von IBM SPSS Collaboration and Deployment Services Deployment Portal können über HTML-Formulare übergeben werden, die in eine Webseite integriert sind. Beispielsweise können einem Benutzer folgende Aktionen über ein Formular ermöglicht werden:

- Auswahl aus einer Liste verfügbarer Berichte
- Auswahl eines Ausgabedateityps
- Angabe von abgefragten Variablen vor der Übergabe der Berichtsanforderung
- Angabe einer ID und eines Kennworts vor dem Ausführen eines Berichts

Das folgende Beispiel referenziert ein benutzerdefiniertes Dialogfeld in der Aktion für ein Formular.

```
<form name='AnalyzeOptions' method='POST' target='Iframe_1'
action='/peb/view/SamplesStatistics/SPD/Simple.spd
?fragment=true&promptstate=1&waitstate=1'>
  <input type='hidden' name='username' value='userA' />
  <input type='hidden' name='password' value='passwordA' />
  <input type='hidden' name='provider' value='Native' />
  <input type='hidden' name='dataset.uri'
value='spsscr:///SamplesStatistics/SAV/multipleResponseData.sav' />
  <input type='hidden' name='allowPivoting' value='false' />

  <input name='PromptParameter1' type='checkbox' value='true' />
  Check the box to select parameter 1

  <br>
  <input type='submit' value='Run Report' />
</form>
```

Verwenden von IBM SPSS Collaboration and Deployment Services Repository zum Speichern benutzerdefinierter Seiten mit relativen Pfaden

Das Repository kann als zentraler Speicherort für alle Dateien für eine benutzerdefinierte Website verwendet werden. Relative oder absolute Pfade können auf der benutzerdefinierten Website verwendet werden, um Elemente, z. B. CSS-Style-Sheets, Bilder, Berichtobjekte von IBM SPSS Collaboration and Deployment Services Deployment Portal und JavaScript, zu verknüpfen.

Sie können beispielsweise einen Ordner mit dem Namen MyWebPage im Repository speichern, der die benutzerdefinierte Webseite MyWebPage.htm und Ressourcen (z. B. Bilder, Style-Sheets und JavaScript-Dateien) enthält. MyWebPage.htm kann **relative** Verweise auf die Ressourcen wie folgende enthalten:

```


<script language="javascript" src="MyJS.js?fragment=true">

</script>

<LINK REL="StyleSheet" HREF="MyStyles.css?fragment=true" TYPE="text/css" MEDIA="screen" />
```

Damit diese relativen Verweise ordnungsgemäß funktionieren, muss über den Parameter `fragment=true` in der URL auf die Webseite zugegriffen werden. Beispiel:

```
http://IhrServer:port/peb/view/MyWebPage/MyWebPage.htm?
username=validUser&password=pass&provider=Native&fragment=true
```

Wenn Sie die Ressourcen für Ihre Website an einer anderen Repository-Position als der Position speichern möchten, an der Ihre Webseite gespeichert ist, können sie von Ihrer Webseite (beispielsweise MyWebPage.htm) mithilfe von **absoluten** Pfaden folgendermaßen referenziert werden:

```


<script language="javascript" src="/peb/view/MyWebPage/js/MyJS.js?fragment=true">
```

```
</script>
```

```
<LINK REL="StyleSheet" HREF="/peb/view/MyWebPage/CSS/MyStyles.css?fragment=true" TYPE="text/css" MEDIA="screen" />
```

Oder Sie verweisen über den vollständigen Hostnamen und Port im Pfad darauf:

```

```

```
<script language="javascript" src="http://IhrServer:8080/peb/view/ MyWebPage/js/MyJS.js?fragment=true">
```

```
</script>
```

```
<LINK REL="StyleSheet" HREF="http://IhrServer:8080/peb/view/ MyWebPage/CSS/MyStyles.css?fragment=true" TYPE="text/css" MEDIA="screen"/>
```

Anmerkung:

Wenn Sie eine HTML-Seite erstellen, die auf veröffentlichte Berichte (.spw) verweist, die in einen HTML-IFRAME eingebettet sind, müssen Sie die Datenschutzeinstellungen des Internet Explorer für Cookies von anderen Anbietern anpassen, um zu vermeiden, dass der Anmeldebildschirm angezeigt wird.

Dieses Szenario tritt nur auf, wenn Sie die .spw-Berichte über eine externe Website aufrufen und dabei IFRAMEs zur Einbettung und Anzeige mehrerer Berichte auf einer Webseite verwendet werden. Beim Zugriff auf die HTML-Seite werden die Berichte ausgeführt und in den IFRAMEs angezeigt. Bei der Einbettung des URL-Parameters von IBM SPSS Collaboration and Deployment Services Deployment Portal in einen IFRAME werden die Cookies als Cookies anderer Anbieter betrachtet. In den Datenschutzeinstellungen des Internet Explorer (Extras/Internetoptionen/Datenschutz) sind Cookies von anderen Anbietern standardmäßig inaktiviert. Dies führt dazu, dass beim Zugriff auf die Webseite der Anmeldebildschirm angezeigt wird (selbst wenn die URL den richtigen Benutzernamen und das richtige Kennwort enthält). Sie können dieses Verhalten in den Datenschutzeinstellungen des Internet Explorer ändern. Fügen Sie den Domännennamen bzw. die IP-Adresse von IBM SPSS Collaboration and Deployment Services mit "Zulassen" zu den verwalteten Sites hinzu. Dadurch wird sichergestellt, dass sämtliche Cookies von dieser Adresse akzeptiert werden und kein Anmeldebildschirm angezeigt wird.

Im Folgenden sehen Sie ein Beispiel für den HTML-Code, der für den Verweis auf einen im Repository veröffentlichten .spw-Bericht verwendet wird.

```
<iframe frameborder=1 src="http://IhrServer:8080/peb/view/jba/accidents.spw?partId=5&fragment=true&username=admin&password=yourpwd&provider=Native" style="WIDTH: 800px;HEIGHT: 280px" name="I1"></iframe>
```

Beachten Sie, dass dieses Problem nur bei Internet Explorer auftritt, nicht bei anderen unterstützten Browsern.

Kapitel 3. IBM SPSS Collaboration and Deployment Services Tag Library

Mit IBM SPSS Collaboration and Deployment Services wird für Administratoren und erfahrene Benutzer, die Beziehungen zwischen Repository-Elementen oder benutzerdefinierte Webseiten (JSP-Seiten) mit Elementen erstellen möchten, die sich gegenseitig mit Werten versorgen können, eine JSP-Tagbibliothek (JavaServer Pages) bereitgestellt.

Die Tagbibliothek bietet folgende grundlegenden Funktionen:

Authentifizierung: Sie können den Benutzer, das Kennwort und den Sicherheitsprovider festlegen und Elemente oder Eingabeaufforderungen freigeben, die auf dieser Seite definiert sind. Für den Zugriff auf Elemente in IBM SPSS Collaboration and Deployment Services Repository und für die Datenquellenauthentifizierung ist eine Authentifizierung erforderlich.

Elemente: Sie können die Definition von Elementen angeben, einschließlich des "Zielcontainers" (<DIV>- oder <IFRAME>-Element). Die Elemente werden für IFRAME-Ziele über eine POST-Anforderung und für DIV-Ziele über eine AJAX-Anforderung (Asynchronous JavaScript und XML) ausgeführt.

Eingabeaufforderungen: Mithilfe von Eingabeaufforderungen können Sie die zum Ausführen der Elemente verwendeten Parameter dynamisch anpassen. Die Eingabeaufforderungsposition ist nur auf eine Position auf der aktuellen Seite beschränkt. Eingabeaufforderungen können entweder benutzerdefiniert oder ein ausgewählter Parameter aus einer vorhandenen Elementdefinition sein.

Verknüpfungsbeziehungen: Sie können Beziehungen zwischen den folgenden Elementen definieren:

- Quellen- und Zielberichtelementen, Job-, Scoring- oder benutzerdefinierten Dialogfeldelementen
- Einer Liste von Eingabeaufforderungen und einem Zielelement. Sowohl die Aktivierungsposition (DIV oder IFRAME) als auch das Timing (ONDEMAND, ONLOAD oder NONE) werden unterstützt.

Die Tagbibliothek setzt sich aus den folgenden wesentlichen Bestandteilen zusammen:

- der öffentlichen JavaScript-API
- benutzerdefinierten Tags und ihren Interaktionen untereinander
- den Tagbibliothekbeans für den Abruf von Datensets

Dieses Dokument beschreibt jede Tagfunktion, die in der JSP-Tagbibliothek verfügbar ist, und enthält einige Verwendungsbeispiele. Wir empfehlen, nach dem Lesen dieses Dokuments die *JSP*-Beispieldateien durchzusehen, die mit der Tagbibliothek bereitgestellt wurden, bevor Sie beginnen, eigene benutzerdefinierte Seiten zu erstellen.

Aktualisieren auf IBM SPSS Collaboration and Deployment Services Tag Library

Beachten Sie, dass bei früheren Versionen von IBM SPSS Collaboration and Deployment Services eine *.tld*-Datei mit der Bezeichnung *reporting-taglib.tld* oder *pasw-taglib.tld* verwendet wurde. Alle vorhandenen JSP-Seiten, die einen dieser Namen verwenden, sollten so aktualisiert werden, dass sie auf *IBMSPSSTaglib.tld* verweisen.

Außerdem sollten Sie prüfen, ob etwaige benutzerdefinierte Seiten, die mit früheren Versionen von IBM SPSS Collaboration and Deployment Services erstellt wurden, in der aktuellen Version erwartungsgemäß arbeiten. Möglicherweise sind einige Abänderungen erforderlich. So machen beispielsweise Änderungen bei der Berichtsverarbeitung Änderungen an Seiten erforderlich, die Berichte beinhalten.

JSP-Architektur

In der Abbildung wird die Architektur für die Verwendung der Tagbibliothek beschrieben. Der Anwendungsserver, auf dem sich IBM SPSS Collaboration and Deployment Services Repository befindetet, enthält eine Servlet-Engine, die die Informationen in den Bibliothektags in Eingaben für Web-Services konvertiert, die in IBM SPSS Collaboration and Deployment Services enthalten sind.

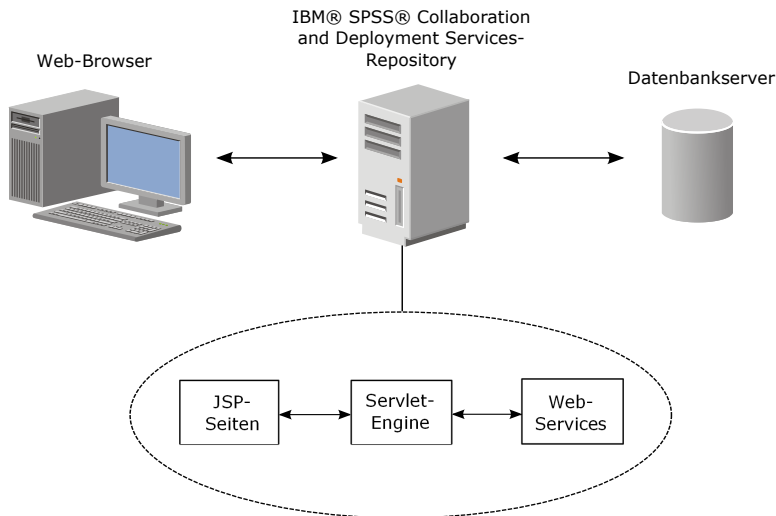


Abbildung 1. JSP-Architektur

Die Ausführung von Elementen über die Tagbibliothek gestaltet sich in der Regel folgendermaßen:

1. Der JSP-Entwickler verwendet benutzerdefinierte Tags, um Berechtigungsnachweise, Elemente und Elementbeziehungen in einer *JSP*-Datei zu definieren, und speichert die Datei auf dem Anwendungsserver, auf dem sich IBM SPSS Collaboration and Deployment Services Repository befindet.
2. Wenn ein Client auf die *JSP*-Seite zugreift, wertet der Server die Tags aus und erzeugt je nach Bedarf XML-Dateninseln oder HTML-Elemente, die von den JavaScript-Komponenten des Frameworks verwendet werden, um Beziehungen zwischen Elementen und Eingabeaufforderungen zu erkennen und zu verwalten.
3. In jedem Tag-Handler werden Validierungen durchgeführt und die entsprechenden Fehlermeldungen werden auf der Seite angezeigt, sodass der Benutzer in jeder Phase der Seitenerstellung über Fehler in Kenntnis gesetzt wird.
4. Ein Servlet unterstützt die Ausführung von Elementen und die Verarbeitung und Rückgabe der Ausgabe.
5. Der entsprechende Web-Service für den Elementtyp wird aufgerufen, um das Element auszuführen und verschiedene Validierungen durchzuführen.

Unterstützte Elemente

Über IBM SPSS Collaboration and Deployment Services Tag Library können verschiedenen Repository-Elemente in *JSP*-Seiten referenziert werden. Bei der Verarbeitung der Seite bestimmt der MIME-Typ des Elements, wie es verarbeitet wird.

Folgende Elemente sind gültig:

- Berichte
- Jobs
- Scoring-Modelle
- Benutzerdefinierte Schnittstellendefinitionen

Berichte

Bei einem Bericht muss das Repository-Element einen der folgenden Dateitypen referenzieren:

- BIRT Report Designer for IBM SPSS-Bericht-Design (*.rptdesign)
- Visualisierungsdefinition (*.viz)

Folgende Eigenschaften sollten bei der Arbeit mit Berichtselementen berücksichtigt werden:

Ausgabe. Ein Berichtselement generiert für gewöhnlich eine einzelne Ausgabe. Visualisierungsberichte generieren jedoch zusätzlich zur Visualisierung eine Bildübersicht. Die Ausgabe für das Element kann, abhängig vom Berichtstyp, in unterschiedlichen Formaten erfolgen. Folgende Formate werden unterstützt:

- HyperText Markup Language (*.html)
- Portable Document Format (*.pdf)
- Berichtsdocument (*.rptdocument)
- HTML Complete (*.htmlc)
- MIME HTML (*.mht)
- Microsoft Word-Dokument (*.doc)
- Microsoft PowerPoint (*.ppt)
- Portable Network Graphic (*.png)
- Enhanced Metafile (*.emf)
- Joint Photographic Experts Group (*.jpeg)

Eingabeaufforderungen. Bei der Verarbeitung wird das Element Werte für alle Variablen anfordern, die im Bericht definiert sind.

Beschränkungen hinsichtlich des Anzeigeorts. Ausgaben vom Typ *.rptdocument können nur in IF-RAME angezeigt werden.

Verknüpfung von Elementen. Berichtselemente können als Quellen für nachfolgende Elemente oder als Ziele für andere Elemente verwendet werden.

Unterstützte Tags. Berichtselemente unterstützen den Tag `outputLocation` nicht. Alle anderen Tags in der Tagbibliothek werden unterstützt.

Das Element kann zusätzliche Informationen enthalten, die die Darstellung der Ausgabe beeinflussen, z. B. Titel des Fensters oder Vorhandensein einer Symbolleiste.

Jobs

Im Fall eines Jobs muss das Repository-Element einen Job in IBM SPSS Collaboration and Deployment Services referenzieren, dessen MIME-Typ `application/x-vnd.spss-prms-job` lautet.

Folgende Eigenschaften sollten bei der Arbeit mit Jobelementen berücksichtigt werden:

Ausgabe. Ein Jobelement kann eine unbegrenzte Anzahl von Ausgaben verschiedener Typen generieren. Welche Ausgabe generiert wird, hängt von den Schritten ab, die im Job enthalten sind.

Eingabeaufforderungen. Bei der Verarbeitung wird das Element Werte für alle Jobparameter anfordern, die im Job definiert sind.

Beschränkungen hinsichtlich des Anzeigeorts. Die Ausgabe der einzelnen Schritte innerhalb des Jobs muss explizit definiert werden.

Verknüpfung von Elementen. Jobelemente können als Ziele für andere Elemente verwendet werden, nicht jedoch als Quellen.

Unterstützte Tags. Jobelemente unterstützen den Tag `actionHandler` nicht. Alle anderen Tags in der Tagbibliothek werden unterstützt.

Scoring-Modelle

Bei einem Scoring-Modell muss das Repository-Element eine für das Scoring definierte Datei referenzieren.

Gültige Dateitypen sind:

- Szenario (*.scn)
- IBM SPSS Modeler-Datenstrom (*.str)
- Predictive Model Markup Language (PMML)
- Definition einer Echtzeitprognoseanwendung

Folgende Eigenschaften sollten bei der Arbeit mit Scoring-Elementen berücksichtigt werden:

Ausgabe. Ein Scoring-Element generiert eine HTML-Ausgabe.

Eingabeaufforderungen. Bei der Verarbeitung kann das Element Werte für Parameter, eine Datendatei, eine Datenproviderdefinition und einen Modellnamen anfordern.

Verknüpfung von Elementen. Scoring-Elemente können als Ziele für andere Elemente verwendet werden, nicht jedoch als Quellen.

Unterstützte Tags. Scoring-Elemente unterstützen die Tags `outputLocation` und `actionHandler` nicht. Alle anderen Tags in der Tagbibliothek werden unterstützt.

Benutzerdefinierte Dialogfelder

Bei einer benutzerdefinierten Webschnittstelle das Repository-Element eine Dialogfelddefinition (*.spd) referenzieren.

Folgende Eigenschaften sollten bei der Arbeit mit benutzerdefinierten Dialogfeldern berücksichtigt werden:

Ausgabe. Ein benutzerdefiniertes Dialogfeld erzeugt eines der folgenden Elemente:

- Eine einzelne Ausgabedatei (*.spw), die einem Frame oder Fenster zugewiesen werden muss.
- Eine HTML-Datei, die einem Frame bzw. Fenster oder einem DIV-Ziel zugewiesen werden kann.

Eingabeaufforderungen. Bei der Verarbeitung wird das Element Werte für alle Eingabeaufforderungen anfordern, die in der Dialogfelddefinition definiert sind. Das Element kann auch Datasets anfordern. In der .spd-Datei definierte Hilfe für Eingabeaufforderungen wird jedoch nicht verwendet. Die Anwendung sollte eigene Hilferenzen enthalten.

Beschränkungen hinsichtlich des Anzeigeorts. Die Ausgabe kann in einem Frame, DIV oder einem neuen Fenster angezeigt werden.

Verknüpfung von Elementen. Dialogfeldelemente können als Ziele für andere Elemente verwendet werden, nicht jedoch als Quellen.

Unterstützte Tags. Dialogfeldelemente unterstützen den Tag `actionHandler` nicht. Alle anderen Tags in der Tagbibliothek werden unterstützt.

Die zur Verwendung in einer URL beschriebenen Webbereitstellungseigenschaften, die ein benutzerdefiniertes Dialogfeld referenziert, können in der Tagbibliothek entweder als im Tag `repositoryItem` verschachtelte Eigenschaften oder mithilfe des Tags `sourceLinkPrompt` angegeben werden.

Die Eigenschaften `dataset.uri` und `dataset.table` sollten immer definiert werden, wobei letztere nur für Datenproviderdefinitionen gilt. Die Eigenschaften `javascript.url`, `javascript.name`, `stylesheet.url` und `stylesheet.name` hingegen werden alle ignoriert. Werte für diese Eigenschaften sollten innerhalb der JSP selbst definiert werden.

Anmerkung:

Für diese Funktionalität sind IBM SPSS Statistics-Adapter in der Umgebung von IBM SPSS Collaboration and Deployment Services erforderlich. Weitere Informationen finden Sie in der Installationsdokumentation zu IBM SPSS Statistics.

Aufbauen einer Anwendung

Jede JSP-Seite in einer benutzerdefinierten Anwendung muss einige Standardaufbauregeln definieren, damit die Tagbibliothek verwendet und fehlerfrei referenziert werden kann.

Die erste Aufbauregel, `page`, legt Eigenschaften für die gesamte Seite fest. Zu diesen Eigenschaften gehören:

- Das Attribut `language`, das die Scripting-Sprache definiert, die von der Seite verwendet wird
- Das Attribut `contentType`, das den MIME-Typ und den Zeichensatz angibt, der für Reaktionen auf Clients verwendet wird
- Das Attribut `session`, das angibt, ob die Tagbibliothek Informationen zur Sitzung speichert

Die zweite Aufbauregel, `taglib`, legt fest, welche Tags von der JSP-Seite verwendet werden. Folgende Eigenschaften sind für diese Aufbauregel definiert:

- Das Attribut `uri`, das den ordnungsgemäßen Pfad zu `IBMSPSSTaglib.tld` angibt
- Das Attribut `prefix`, das einen Bereich für die Tags definiert

Beachten Sie, dass bei früheren Versionen von IBM SPSS Collaboration and Deployment Services eine `.tld`-Datei mit der Bezeichnung `reporting-taglib.tld` oder `pasw-taglib.tld` verwendet wurde. Alle vorhandenen JSP-Seiten, die einen dieser Namen verwenden, sollten so aktualisiert werden, dass sie auf `IBMSPSSTaglib.tld` verweisen.

Im folgenden Beispiel wird die Aufbauregel `page` verwendet, um den Inhaltstyp als `text/html` mit dem Zeichensatz UTF-8, der Scripting-Sprache Java und dem `session`-Objekt "true" zu definieren. Die `taglib`-Aufbauregel legt den Speicherort der Berichtsdatei `.tld` und ein Präfix `r` für alle darin definierten Tags fest.

```
<%@ page contentType="text/html;charset=utf-8"
    language="java" session="true" %>

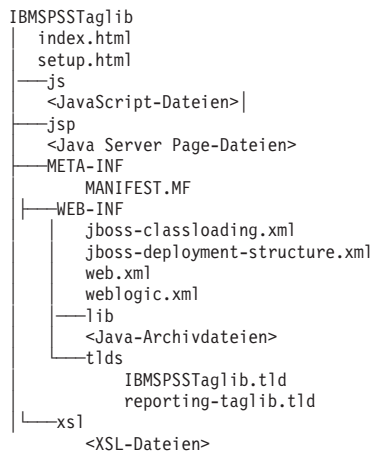
<%@ taglib uri="/WEB-INF/tlds/IBMSPSSTaglib.tld" prefix="r" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    http://www.w3.org/TR/html4/loose.dtd">

<html>
<!-- Rest von HTML / JSP ist hier -->
</html>
```

Um Ihre Anwendung in die Produktion zu geben, sollten Sie die Erstellung einer Archivdatei für die Webanwendung (`.war`) mit den `JSP`-Dateien planen und sie als separate Webanwendung auf Ihrem Anwendungsserver bereitstellen. Dies ist die bevorzugte Methode.

Die Struktur des erweiterten Anwendungsarchivs der Beispieltagbibliothek "Reporting" (*IBMSPSSTaglib.tld.war*) in der Standardinstallation von IBM SPSS Collaboration and Deployment Services lautet beispielsweise folgendermaßen:



Beachten Sie, dass die TLD-Datei (Tag Library Descriptions) und die TLD-Bibliotheken (*JAR-Dateien*) in der bereitgestellten *WAR-Datei* enthalten sind. Die TLD-Datei wird auch in der Deskriptordatei der Anwendung (*web.xml*) referenziert:

```

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.3/EN"
"http://java.sun.com/dtd/web-app_2_3.dtd">
<web-app>
  <servlet>
    <servlet-name>ReportingTaglibServlet</servlet-name>
    <display-name>
      Servlet responsible for fulfilling all requests from
      reporting taglibs
    </display-name>
    <servlet-class>
      com.spss.report.taglib.servlet.ReportingTaglibServlet
    </servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name>ReportingTaglibServlet</servlet-name>
    <url-pattern>/reportingTaglib/*</url-pattern>
  </servlet-mapping>

  <servlet-mapping>
    <servlet-name>ReportingTaglibServlet</servlet-name>
    <url-pattern>/tagLib/*</url-pattern>
  </servlet-mapping>

  <error-page>
    <error-code>500</error-code>
    <location>/jsp/error/error.jsp</location>
  </error-page>
</web-app>

```

Der Deskriptor der Anwendung gibt ebenfalls an, dass das Servlet *ReportingTaglibServlet* den URL-Mustern */taglib* und */reportingTagLib* zugewiesen wird und beide URLs das Servlet aufrufen können. Die Java-Klasse des Servlets lautet *com.spss.report.taglib.servlet.ReportingTaglibServlet*. Der optionale Single Sign-on wird durch einen Servletfilter *Common Authentication Filter* aktiviert, der die Klasse *com.spss.report.taglib.filter.AuthFilter* verwendet und dem Servlet über eine URL zugewiesen ist. Der Filter wird über Parameter zur Verschlüsselung und für die Single-Sign-on-Adapterklasse initialisiert.

Weitere Informationen und Anforderungen für *WAR-Dateien* finden Sie in der Dokumentation zu Ihrem Anwendungsserver.

Wichtig! Für *WAR-Anwendungsdateien*, die nicht von Installationscripts von IBM SPSS Collaboration and Deployment Services oder IBM SPSS Collaboration and Deployment Services Package Manager bereitgestellt werden, wie Tagbibliothek- oder benutzerdefinierte Anwendungen, muss evtl. die Klassenreihenfolge geändert werden. Beispielsweise muss sie für Tagbibliotheksanwendungen für Berichte

und Scoring unter WebSphere auf *Klassen mit Anwendungsklassenlader zuerst geladen* und die WAR-Klassenladerrichtlinie auf *Einzelner Klassenlader für Anwendung* gesetzt sein.

Implementierungsdetails

Benutzer, die benutzerdefinierte JSP-Seiten erstellen, sollten sich über folgende Punkte im Klaren sein:

- Jedes Mal, wenn der Server anhält, gehen JSP-Dateien, die direkt im Verzeichnis *tmp/deploy* gespeichert sind, verloren. Um Ihre JSP-Dateien beizubehalten, speichern Sie Sicherungskopien auf einer lokalen Festplatte und kopieren sie nach jedem Neustart wieder auf den Server. Produktionsanwendungen sollten in WAR-Dateien gepackt werden.
- Bei Berichten, die Bilder oder Diagramme enthalten, kombiniert das .mht-Ausgabeformat alle Bilder/Diagramme und HTML-Elemente in einer einzigen Datei, die nur mit Internet Explorer kompatibel ist.
- Der Server validiert alle Tags soweit wie möglich vorab und schreibt eventuell festgestellte Fehlernachrichten an die HTTP-Antwort. Das unterstützt den JSP-Entwickler bei der Behebung von Problemen mit der Erstellung einer Seite. Es werden beispielsweise die folgenden Validierungen durchgeführt:
 - Prüfung, ob alle erforderlichen Berechtigungsnachweise definiert sind
 - Prüfung der Berechtigungsnachweise
 - Prüfung, ob Berichtsparameter vorhanden sind
 - Prüfung, ob für ein Berichtsobjekt Spaltennamen vorhanden sind
 - Prüfung, ob die Verknüpfung zwischen Elementen logisch schlüssig ist
- Für die Tags ist ein Repository-Server erforderlich.
- Die Tagbibliothek unterstützt Verknüpfungen zwischen Eingabeaufforderungen und Elementen, zwischen dem Klicken auf eine Zeile und Zielelementen, zwischen Berichten und Drilldown-Berichten in demselben Frame und zwischen Eingabeaufforderungs-/Tabellenzeilen und Zielelementen, die in separaten Fenstern geöffnet sind.
- Verknüpfungsverhalten wird dem Benutzer nicht angezeigt und über `sourceLinkRepositoryItem`- oder `sourceLinkPrompt`-JSP-Tags definiert. Der Benutzer muss die Technologie hinter der JSP-Tagverwendung nicht verstehen.
- Alle Zielelemente müssen mit Parametern vordefiniert werden, um die weitergegebenen Parameter zu erhalten.
- Für benutzerdefinierte Dialogfelder definiert das Standard-CSS Standardstile, die bei Verwendung des Tags `repositoryItem` automatisch mit einbezogen werden. Um diese Stile zu überschreiben, müssen Sie nach dem Tag `repositoryItem` ein benutzerdefiniertes Style-Sheet einfügen. Beispiel:

```
<r:repositoryItem name="sample" inputURI="spsscr:///myDialog.spd"
  ...weitere Angaben...
</r:repositoryItem>

<link rel="stylesheet" type="text/css" href="MyStyles.css">
```

Öffentliche JavaScript-API

Das Framework stellt JavaScript-Funktionen zur Verarbeitung von Repository-Elementen, für den Abruf von mit Lesezeichen versehenen Berichtswerten und von Werten kaskadierender Eingabeaufforderungen bereit.

Funktion "runRepositoryItem"

Die öffentliche JavaScript-Funktion `runRepositoryItem` wird vom Framework für das Ausführen eines Elements bereitgestellt. Mit ihr kann der Entwickler ein Element ausführen, indem er dieses JavaScript mit einem Ereignishandler verknüpft, und ein Element über Eingabeaufforderungen aktivieren.

Wenn ein Element jedoch direkt über diese Funktion ausgeführt wird, wird die normale Eingabeaufforderungsvalidierung umgangen. Die Anwendung ist dafür zuständig, Parameter zu validieren, bevor die Funktion aufgerufen wird.

Die Funktion nimmt folgende drei Argumente an.

- Eine Zeichenfolge für den Namen eines Elements, das ausgeführt werden soll. Der Name muss über das Attribut `name` des Tags `repositoryItem` definiert worden sein.
- Ein Array von Datenwerten, die bei der Ausführung des Elements als Parameterwerte verwendet werden sollen. Das Array besitzt die folgende Struktur:

```
var thisVar = new Object();
thisVar.value = "param_value";
thisVar.columnName = "param_name";
var linkedData = new Array(thisVar);
```

- Einen optionalen Parameter, der die Zielposition für die Elementausgabe angibt. Er folgt denselben Regeln wie das Attribut `location` des Tags `repositoryItem`. Dabei kann es sich um die ID eines DIV-Elements, den Namen eines IFRAME- oder FRAME-, **NEW-* oder **windowName-*Elements handeln. Wenn dieser Parameter nicht angegeben wird, wird der Standardspeicherort aus `repositoryItem` verwendet.

Das Datenwertarray kann folgendermaßen definiert werden:

- Als NULL (oder weggelassen). In diesem Fall werden die erforderlichen Werte über `sourceLinkPrompt`-Tags abgerufen, die für das Element definiert sind.
- Als Verknüpfungsdaten aus `actionHandler`. Die `actionHandler`-Tags definieren Funktionen, die aufgerufen werden sollen, und die Parameterwerte, die im Rahmen des Ereignisses erfasst werden. Diese Parameter können direkt an die `runRepositoryItem`-API weitergegeben werden.
- Als benutzerdefiniert. Das JavaScript, das die API `runRepositoryItem` aufruft, kann die Werte des Arrays bei Bedarf definieren. `columnName` ist der Name der Spalte, die im Tag `sourceLinkVariable` definiert wird. Der Wert ist der Wert, der an den Parameter weitergegeben wird.

Funktion "getBookmarkedValues"

Die Funktion `getBookmarkedValues` ruft die Werte von Zellen ab, die in einem Bericht von BIRT Report Designer for IBM SPSS mit einem Lesezeichen versehen wurden.

Das Attribut `"id"` der Zelle sollte auf den Lesezeichenwert eingestellt werden. Diese Funktion kann zur Verknüpfung komplexer Berichte von BIRT Report Designer for IBM SPSS mit Kreuztabellen und verschachtelten Tabellen verwendet werden.

Die Funktion `getBookmarkedValues` nimmt die folgenden Argumente an:

- Einen übergeordneten Knoten im DOM des Berichts, den die Funktion benötigt, um die Zellenwerte abrufen zu können, die den im Array des Lesezeichens angegebenen Elementen entsprechen.
- Ein Array mit Lesezeichen, die im Bericht definiert sind, dessen Werte benötigt werden. Beispiel: `["bookid1","bookid2"]`.

Die Funktion gibt ein Array von Datenwerten aus, die bei der Ausführung des Berichts als Parameterwerte verwendet werden sollen. Das Array besitzt die folgende Struktur:

```
var thisVar = new Object();
thisVar.value = "cell_value";
thisVar.columnName = "bookmark";
```

`columnName` ist der Name des Lesezeichens. `value` ist der Wert der angegebenen Zelle, die mit einem Lesezeichen versehen wird.

Funktion "retrievePromptValues"

Die Funktion `retrievePromptValues` sollte aufgerufen werden, wenn Parameter mit benutzerdefinierten Steuerelementen verwendet werden. Sie unterstützt sowohl kaskadierende als auch nicht kaskadierende Eingabeaufforderungen.

Rufen Sie diese Funktion im Textkörper des Handlers `onLoad` auf, um die Anfangswerte der Eingabeaufforderung zu laden (oder der übergeordneten Eingabeaufforderung bei kaskadierenden Eingabeaufforderungen). Rufen Sie diese Funktion im Handler `onChange` des für die Definition des Weitergabeparameters

verwendeten Steuerelements auf. Die Funktion ruft vom Server die Eingabeaufforderungswerte zum Auffüllen der Parametersteuerelemente mit aktualisierten Werten abhängig vom ausgewählten übergeordneten Parameterwert ab.

Die Funktion `retrievePromptValues` nimmt die folgenden Argumente an:

- Eine Zeichenfolge mit dem Namen des Berichts, der die Definitionen der Weitergabeparameter enthält. Der Name muss als `name`-Attribut eines `repositoryItem`-Tags definiert worden sein.
- Eine Zeichenfolge, die dem Namen des Parameters im Bericht entspricht. Für Weitergabeparameter ist diese Zeichenfolge der Name der Weitergabeparametergruppe. Die Weitergabeparametergruppe muss im Bericht vorhanden sein.
- Eine benutzerdefinierte Funktion, die ein Array von Werten akzeptiert und Text für die neuen Optionen anzeigt. Das Array kann "Null" sein. In diesem Fall sollte die Funktion das Steuerelement löschen. Diese Funktion wird über `retrieveCascadingPromptValues` aufgerufen, um die Parametersteuerelemente mit neuen Werten zu füllen.

```
function callback(options) {
  // Logik zum Löschen des Steuerelements
  // Logik zum Hinzufügen von Werten und Anzeigen von Text für Steuerelemente
  for(var i = 0; i < options.length; i++) {
    control.value = options[i].value;
    Display Text for control = options[i].displayText;
  }
}
```

- Ein Array der ausgewählten vorangehenden Werte in der Weitergabeparametergruppe. Dieses Array wird nur für Weitergabeparameter benötigt und sollte für einen Nicht-Weitergabeparameter ausgelassen werden. Die Parameter müssen in sequenzieller Reihenfolge sein. Um die Liste des übergeordneten Weitergabeparameters abzurufen, geben Sie die vorangehenden Werte an:

```
var precedingvals= new Array();
```

Das Array mit den vorangehenden Werten besitzt die folgende Struktur: Beispielsweise zum Abrufen einer Liste der Städte in Minnesota (MN):

```
precedingvals= new Array();
precedingvals[0]= "USA";
precedingvals[1]="MN";
```

IBM SPSS Collaboration and Deployment Services Tag Library - Tagreferenz

Die verschiedenen Tags in IBM SPSS Collaboration and Deployment Services Tag Library sind voneinander abhängig und zu Validierungszwecken muss sichergestellt sein, dass die Referenzen korrekt eingehalten werden. Außerdem müssen die Tags in der korrekten Reihenfolge definiert werden.

In den folgenden Abschnitten wird jeder verfügbare Tag ausführlich beschrieben.

Diese Tagbibliothek basiert auf JSP 1.2.

Tag "credential"

Der Tag `credential` definiert sowohl die Anmeldeberechtigungsanfrage für eine Datenquelle als auch für ein Repository.

Der Tag "credential" wird für alle auf der Seite definierten Elemente und Eingabeaufforderungen nach Namen referenziert. Er sollte vor allen anderen Tags definiert werden, die den Tag "credential" eventuell referenzieren. Bei normaler Verwendung wäre es der erste in der JSP referenzierte Tag.

Der Tag `credential` kann `properties`-Elemente enthalten. Im Fall von JDE-Datenquellen (J. D. Edwards) sieht der Tag `credential` beispielsweise folgendermaßen aus:

```

<credential>
  <properties>
    <property name="JDE_LIBRARY_LIST_SELECTED" value="liblist_name"/>
  </properties>
</credential>

```

Tabelle 4. Attribute für den Tag "credential".

Name	Erforderlich	Beschreibung
name	true	<p>Entweder ein interner Name für die Berechtigungsnachweise für IBM SPSS Collaboration and Deployment Services Repository oder der Name einer in einem Repository-Objekt verwendeten Datenquelle. Wird verwendet, um Elemente und Eingabeaufforderungen mit diesen Berechtigungsnachweisen zu verknüpfen und für alle erforderlichen Anmeldungen von referenzierten Elementen bei Datenquellen.</p> <ul style="list-style-type: none"> • Für Berechtigungsnachweise für ein Repository muss dieser Name mit dem im Attribut repositoryCredentialName des Tags repositoryItem angegebenen Namen übereinstimmen. • Für Datenbankberechtigungs-nachweise muss der Name dem Namen der Datenquelle entsprechen, da er vom Element referenziert wird, das diese Datenquelle verwendet. <p>Dieser Name wird verwendet, um die Berechtigungsnachweise in einer Sitzungsvariablen zu speichern. Jeder Tag "credential" muss einen eindeutigen Namen besitzen.</p>
useSSO	false	<p>Gibt an, ob Single-Sign-on-Berechtigungsnachweise für Kerberos verwendet werden sollen. Falls dieses Attribut auf "true" eingestellt ist, dürfen die Attribute username, password und provider nicht angegeben werden.</p> <p>Wenn Single Sign-on verwendet wird, muss der Authentifizierungsfilter in der Datei web.xml konfiguriert werden.</p>

Tabelle 4. Attribute für den Tag "credential" (Forts.).

Name	Erforderlich	Beschreibung
credentialDefinitionName	false	Der Name eines Berechtigungsnachweises, der im Repository als Ressource definiert ist. Falls dieser Wert angegeben ist, müssen die Attribute username, password und provider nicht angegeben werden, da die Berechtigungsnachweisressource diese Informationen bereits enthält.
provider	false	<p>Für Berechtigungsnachweise für ein Repository ist dies der optionale Name eines Sicherheitsanbieters. Gültige Werte sind:</p> <ul style="list-style-type: none"> • <i>Native</i> für den integrierten Anbieter. • <i>AD_<Name>/<Domäne></i> für Active Directory, wobei <Name> dem Namen des Sicherheitsproviders im System und <Domäne> dem DNS-Namespace entspricht • <i>ADL_<Name>/<Domäne></i> für Active Directory mit lokaler Überschreibung, wobei <Name> dem Namen des Sicherheitsproviders im System und <Domäne> dem DNS-Namespace entspricht • <i>iSeries</i> für IBM i • <i>LDAP_<Name></i> für OpenLDAP, wobei <Name> dem Namen des Sicherheitsproviders im System entspricht <p>Falls kein Name angegeben ist, wird der integrierte Repository-Sicherheitsanbieter verwendet. Dieses Attribut wird für Berechtigungsnachweise für Datenbanken ignoriert.</p>
username	false	Der Benutzername, der für die Authentifizierung verwendet wird.
password	false	Das Kennwort für den angegebenen Benutzernamen. Das Kennwort wird intern von der Tagbibliothek verwendet. Es wird NICHT in das JSP-Ergebnis geschrieben.

Tagverschachtelung

Keine

Erwartete Ausgabe

Keine. Dieser Tag enthält Authentifizierungsinformationen. Der Tag liefert keine Ausgabe, sondern nimmt die Berechtigungsnachweise mithilfe des Attributs `name` als Schlüssel zur späteren Verwendung mit einem Tag "report" oder "prompt" in den Cache auf.

Verwendungsbeispiel

In folgendem Beispiel sind drei credential-Tags angegeben: Der erste dient zum Zugriff auf IBM SPSS Collaboration and Deployment Services Repository über einen vorgegebenen Benutzernamen und ein vorgegebenes Kennwort. Der Wert *Native* für *provider* zeigt an, dass Benutzernamen und Kennwort für die Validierung im nativen (bzw. lokalen) Sicherheitsprovider definiert sind. Der zweite Tag "credential" ermöglicht Single Sign-on für IBM SPSS Collaboration and Deployment Services Repository unter Verwendung der zuvor authentifizierten Berechtigungsnachweise des Benutzers. Der dritte Tag "credential" gilt für eine Datenquelle mit der Bezeichnung *Northwind*.

```
<r:credential name="repositoryCredential" provider="Native"
  username='admin' password='password' />
<r:credential name="repositorySSO" useSSO="true" />
<r:credential name="Northwind" username='sa' password='sa' />
```

Tag "repositoryItem"

Der Tag `repositoryItem` ist der wichtigste Tag zur Definition von Repository-Elementdefinitionen, die von der Anwendung verwendet werden.

Der Tag `repositoryItem` kann Berichte (Berichte von BIRT Report Designer for IBM SPSS oder Visualisierungsberichte), Jobs, Scoring-Elemente oder SPD-Dateien referenzieren. Die Repository-Elemente können direkt ausgeführt, zur Bereitstellung von Eingabeaufforderungen verwendet oder programmgesteuert ausgeführt werden.

`sourceLinkPrompt`- und `sourceLinkRepositoryItem`-Tags sollten wie folgt im Tag `repositoryItem` verschachtelt sein.

- Verwenden Sie einen verschachtelten Tag `sourceLinkRepositoryItem`, falls dieses Element ausgeführt wird, wenn der Benutzer auf ein anderes Element klickt.
- Verwenden Sie den Tag `sourceLinkPrompt`, wenn die Parameterwerte von auf der Seite oder direkt im Element definierten Eingabeaufforderungen stammen.

Optional können Sie zusätzliche Eigenschaften angeben, die für einen bestimmten Repository-Elementtyp spezifisch sind. Damit sie im Firefox-Browser verwendet werden können, müssen die Eigenschaftsnamen in Kleinbuchstaben stehen. Diese Eigenschaftswerte werden an die URL übergeben, damit das Repository-Element ausgeführt wird. Die Eigenschaften werden als verschachtelter XML-Block angegeben.

Table 5. Attribute für den Tag "repositoryItem".

Name	Erforderlich	Beschreibung
name	true	Definiert einen eindeutigen Namen für das Element. Der Name kann danach von anderen Tags oder über die JavaScript-API <code>runRepositoryItem()</code> referenziert werden.

Tabelle 5. Attribute für den Tag "repositoryItem" (Forts.).

Name	Erforderlich	Beschreibung
inputURI	true	<p>Die für das Rendering der Berichtsausgabe zu verwendende Elementdefinition. Dieser Wert muss einen URI angeben, der zur Suche nach der Elementdefinition verwendet werden kann. Die folgenden URI-Schemata werden unterstützt:</p> <ul style="list-style-type: none"> • <i>file</i>: Referenziert eine bestimmte Datei auf dem Anwendungsserver oder an einer Netzdateiposition. • <i>SPSSCR</i>: Referenziert eine Datei in IBM SPSS Collaboration and Deployment Services Repository. Dieses Schema ermöglicht die Referenzierung von Dateien durch IDs oder hierarchische Pfade im Repository. Es können spezielle Versionskennzeichnungen angegeben werden. Falls keine Version oder Beschriftung angegeben wird, wird die letzte Version verwendet. • <i>scoring</i>: Referenziert eine Modellkonfiguration aus dem Repository. Scoring-Konfigurationen werden nach dem Namen aus den Tagbibliotheken referenziert. Falls eine Scoring-Konfiguration umbenannt wird, muss auch die Tagbibliotheksreferenz geändert werden.
activate	true	<p>Gibt an, wann das Element aktiviert wird. Zu den Optionen gehören:</p> <ul style="list-style-type: none"> • <i>ONDEMAND</i>: Führt das Element aus, wenn es durch Klicken auf eine Zeile für einen Quellenbericht aktiviert wird. • <i>ONLOAD</i>: Führt das Element aus, wenn die Seite zum ersten Mal geladen wird. • <i>NONE</i>: Das Element wird nicht automatisch ausgeführt. In diesem Fall wird das Element zur Bereitstellung von Eingabeaufforderungen oder von Werten für Eingabeaufforderungen verwendet. <p>Unabhängig von der Aktivierungseinstellung kann jeder Bericht durch Verwendung der öffentlichen JavaScript-API <code>runRepositoryItem()</code> programmgesteuert ausgeführt werden.</p>

Tabelle 5. Attribute für den Tag "repositoryItem" (Forts.).

Name	Erforderlich	Beschreibung
location	false	<p>Das Ziel für die Ausgabe des ausgeführten Elements. Die Verwendung variiert etwas abhängig vom Zieltyp.</p> <ul style="list-style-type: none"> • Bei DIV-Zielen sollte das Attribut "location" die ID des DIV-Tags angeben, in dem die Ausgabe platziert werden soll. • Bei IFRAME-Zielen muss das Attribut "location" den Namen des Frames angeben. • Um die Ausgabe in einem neuen Fenster zu öffnen, geben Sie als Speicherort *NEW an. • Um die Ausgabe an ein benanntes Fenster umzuleiten, verwenden Sie einen Stern (*) gefolgt vom Fensternamen. Beispielsweise öffnet *MEINFENSTER ein neues Fenster mit dem Namen <i>MEINFENSTER</i> und verwendet dieses Fenster erneut bei jeder Aktivierung der Verknüpfung.
repositoryCredentialName	true	<p>Der Name der zu verwendenden Berechtigungsnachweise, wenn über das Repository auf das Element zugegriffen wird und wenn das Element über den Reporting-Service ausgeführt wird. Die Berechtigungsnachweise sollten zuvor über den Tag <i>credential</i> definiert worden sein.</p>

Tabelle 5. Attribute für den Tag "repositoryItem" (Forts.).

Name	Erforderlich	Beschreibung
outputType	false	<p>Der zu erstellende Ausgabotyp. Die unterstützten Ausgabetypen variieren je nach Elementtyp. Normalerweise erfolgt die Ausgabe im HTML- oder PNG-Format. Es sind jedoch auch folgende Optionen möglich:</p> <p>Berichte von BIRT Report Designer for IBM SPSS: <i>HTML, PDF, RPTDocument, PowerPoint, Word-Dokument, HTMLC</i></p> <p>Visualisierungsberichte: <i>PNG, EMF, JPEG, HTML</i></p> <p>Ohne Angabe erfolgt die Ausgabe standardmäßig im HTML-Format (oder im PNG-Format für Visualisierungsberichte). Wenn ein Bericht von BIRT Report Designer for IBM SPSS als Verknüpfungsquelle verwendet wird, wird das Attribut outputType ignoriert und eine Ausgabe im HTML-Format erstellt, da die anderen Ausgabetypen keine Verknüpfungen unterstützen.</p> <p>Um Berichte mit dem Viewer anzeigen zu können, geben Sie den Typ <i>RPTDocument</i> an. Für diesen Typ muss das Ziel ein IFRAME-Ziel sein.</p>
showTitle	false	<p>Gibt an, ob die Titelleiste des BIRT-Viewers angezeigt werden soll. Geben Sie als Wert entweder "true" oder "false" an. Diese Einstellung gilt nur, wenn für das Attribut outputType das Format <i>RPTDocument</i> angegeben ist, das im Berichtsviewer angezeigt wird. Der Standardwert lautet true.</p>
title	false	<p>Gibt den Titel für den BIRT-Viewer an. Diese Einstellung gilt nur, wenn für das Attribut outputType das Format <i>RPTDocument</i> angegeben ist, das im Berichtsviewer angezeigt wird. Falls kein Titel angegeben ist, wird der Standardtitel angezeigt.</p>

Tabelle 5. Attribute für den Tag "repositoryItem" (Forts.).

Name	Erforderlich	Beschreibung
showToolBar	false	Gibt an, ob die Symbolleiste des BIRT-Viewers angezeigt werden soll. Geben Sie als Wert entweder "true" oder "false" an. Diese Einstellung gilt nur, wenn für das Attribut <i>outputType</i> das Format <i>RPTDocument</i> angegeben ist, das im Berichtsviewer angezeigt wird. Der Standardwert lautet true.
showNavigationBar	false	Gibt an, ob die Navigationsleiste des BIRT-Viewers angezeigt werden soll. Geben Sie als Wert entweder "true" oder "false" an. Diese Einstellung gilt nur, wenn für das Attribut <i>outputType</i> das Format <i>RPTDocument</i> angegeben ist, das im Berichtsviewer angezeigt wird. Der Standardwert lautet true.
width	false	Die Breite des Ausgabebilds. Die Breite muss größer als 0 sein und zusammen mit der Höhe angegeben werden. Falls keine Breite angegeben ist, werden die Standardbreite und -höhe verwendet.
height	false	Die zu verwendende Höhe, falls es sich bei der Ausgabe um ein Bild handelt. Ohne Angabe einer Breite hat diese Einstellung keinerlei Auswirkungen. Der Wert muss größer als 0 sein.

Tagverschachtelung

Dieser Tag kann einen Tag `sourceLinkRepositoryItem` und mehrere `sourceLinkPrompt`- und `outputLocation`-Tags enthalten.

Verwendungsbeispiel

Das folgende Beispiel definiert ein Element mit dem Namen *AllCountries* für einen in IBM SPSS Collaboration and Deployment Services Repository gespeicherten Bericht von BIRT Report Designer for IBM SPSS.

```
<r:repositoryItem name="AllCountries"
  inputURI="spsscr:///SampleReports/BIRT/CountrySales.rptdesign"
  repositoryCredentialName="repositoryCredential"
  outputType="HTML" width="400" height="300"
  activate="ONLOAD" location="ReportDIV">
</r:repositoryItem>
```

Um Parameterwerte für ein Element anzufordern, geben Sie einen Tag `sourceLinkPrompt` mit an. Im folgenden Beispiel wird mithilfe der JavaScript-Funktion `getValue` ein Wert für den Parameter `ShipCountry` abgerufen.


```

<r:repositoryItem name="CountrySales"
  inputURI="spsscr:///SampleReports/BIRT/CountrySalesByCity.rptdesign"
  repositoryCredentialName="repositoryCredential"
  outputType="HTML" activate="ONDEMAND" location="ReportDIV">
  <r:sourceLinkPrompt targetNameParameter="ShipCountry"
    getValueJSFunction="getValue('IDFilter')"/>
</r:repositoryItem>

```

Um ein zweites Element als Reaktion auf eine Benutzeraktion auszuführen, geben Sie außerdem einen Tag `sourceLinkRepositoryItem` an. Im folgenden Beispiel wird der Visualisierungsbericht `CitiesBarChart.viz` als Reaktion auf eine Aktion im Quellenbericht `AllCountries` ausgeführt.

```

<r:repositoryItem name="CityDetails"
  inputURI="spsscr:///SampleReports/Vis/CitiesBarChart.viz"
  repositoryCredentialName="repositoryCredential"
  outputType="png" width="400" height="300"
  activate="ONDEMAND" location="SecondReportDIV">
  <r:sourceLinkRepositoryItem sourceReportName="AllCountries">
    <r:sourceLinkVariable columnName="ShipCountry"
      targetNameParameter="ShipCountry" />
  </r:sourceLinkRepositoryItem>
</r:repositoryItem>

```

Tag "repositoryItemPrompt"

Der Tag `repositoryItemPrompt` generiert den HTML-Code für eine Eingabeaufforderungsvariable, die im referenzierten Element definiert ist.

Das Element, das die Eingabeaufforderung referenziert, muss über den Tag `repositoryItem` definiert werden, bevor dieser Tag verwendet werden kann. Verwenden Sie diesen Tag, wenn in Ihrer Anwendung Eingabeaufforderungen wie die in IBM SPSS Collaboration and Deployment Services Deployment Portal verwendet werden sollen.

Dieser Tag generiert die HTML-Eingabeaufforderung an dem Ort, an dem der Tag verwendet wird. Der Tag muss einem bestimmten Parameter eines Elements zugeordnet sein, um von Nutzen zu sein. Die Zuordnung zu Parametern geschieht über den Tag `sourceLinkPrompt`, wobei der `promptID`-Wert von `sourceLinkPrompt` mit dem `promptID`-Wert dieses Tags übereinstimmen muss.

Tabelle 6. Attribute für den Tag "repositoryItemPrompt".

Name	Erforderlich	Beschreibung
<code>promptId</code>	false	Eine eindeutige ID, die über das Attribut <code>promptId</code> des Tags <code>sourceLinkPrompt</code> referenziert werden kann.
<code>repositoryItemName</code>	true	Ein Verweis auf den Namen des Elements wie im Attribut <code>name</code> des Tags <code>repositoryItem</code> definiert.
<code>parameterName</code>	false	Name der Eingabeaufforderungsvariable wie im Element definiert.

Tagverschachtelung

Keine

Erwartete Ausgabe

Ein HTML-Element, das es dem Benutzer ermöglicht, je nach `promptType`, der als `parameterName` ausgewählt wurde, persönliche Werte auszuwählen oder einzugeben. Der Tag `repositoryItemPrompt` unter-

stützt alle Parameter, die in IBM SPSS Collaboration and Deployment Services Deployment Portal unterstützt werden. Daher werden alle Eingabeaufforderungstypen unterstützt und das entsprechende HTML-Element wird erzeugt.

Verwendungsbeispiel

In folgendem Beispiel wird die Eingabe eines Werts für den Parameter *EmployeeID* im Bericht *Employees* angefordert.

```
<repositoryItem name="Employees"
  inputURI="file:///d:/yourDS/ReportTaglib/Employees.dbq"
  repositoryCredentialName="localhost" activate="NONE" />

<repositoryItemPrompt promptId="EmployeeIdPrompt"
  repositoryItemName="Employees" parameterName="EmployeeID" />
```

Tag "report"

Dieser Tag wird nicht weiter unterstützt. Verwenden Sie stattdessen den Tag `repositoryItem`.

Tag "reportPrompt"

Dieser Tag wird nicht weiter unterstützt. Verwenden Sie stattdessen den Tag `repositoryItemPrompt`.

Tag "outputLocation"

Dieser Tag verknüpft die generierte Ausgabe, die im Repository vorhanden ist, mit der Position auf der Seite, an der die Ausgabe angezeigt wird. Bei Ausführung des Elements wird die Ausgabe aus dem Repository abgerufen und an der angegebenen Zielposition auf der Seite angezeigt.

Dieser Tag muss immer in einem Tag `repositoryItem` verschachtelt sein.

Tabelle 7. Attribute für den Tag "outputLocation".

Name	Erforderlich	Beschreibung
outputId	false	Dies ist der Pfad der im Repository vorhandenen Ausgabe. Für benutzerdefinierte Dialogfelder sollte dieses Attribut ausgelassen werden. Die Ausgabe aus der Ausführung der Syntax wird automatisch erkannt.

Tabelle 7. Attribute für den Tag "outputLocation" (Forts.).

Name	Erforderlich	Beschreibung
location	true	<p>Dieses Attribut gibt an, wo auf der Seite die Ausgabe eingefügt werden soll.</p> <ul style="list-style-type: none"> • Bei DIV-Zielen sollte das Attribut "location" die ID des DIV-Tags angeben, in dem die Ausgabe platziert werden soll. • Bei IFRAME-Zielen muss das Attribut "location" den Namen des Frames angeben. • Um die Berichtsausgabe in einem neuen Fenster zu öffnen, geben Sie als Speicherort *NEW an. • Um die Ausgabe an ein benanntes Fenster umzuleiten, verwenden Sie einen Stern (*) gefolgt vom Fensternamen. Beispielsweise öffnet *MEINEBERICHTE ein neues Fenster mit dem Namen <i>MEINEBERICHTE</i> und verwendet diesen Link erneut bei jeder Aktivierung der Verknüpfung. <p>Ausgaben im HTML-Format können ein DIV-Ziel haben. Alle anderen Ausgaben sollten IFRAME-Ziele oder ein Fenster zum Ziel haben.</p>
partId	false	<p>Dies wird verwendet, um den spezifischen Teil oder das spezifische Element der Ausgabe des SPW-Archivs zu kennzeichnen.</p> <p>Für diese Funktionalität sind IBM SPSS Statistics-Adapter in der Umgebung von IBM SPSS Collaboration and Deployment Services erforderlich. Weitere Informationen finden Sie in der Installationsdokumentation zu IBM SPSS Statistics.</p>

Tagverschachtelung

Keine

Verwendungsbeispiel

Im folgenden Beispiel wird mithilfe des IFRAME-Tags *ChartFRAME* eine Ausgabeposition für eine Tabelle angegeben, die in IBM SPSS Collaboration and Deployment Services Repository gespeichert ist.

```
<outputLocation outputId="spsscr:///output/output_chart.png"
  location="ChartFRAME"/>
```

Falls die Attributwerte von Parameterwerten abhängig sein, verwenden Sie den Tag `sourceLinkPrompt`, um Übereinstimmungen für die Parameter zu definieren. Falls eine Übereinstimmung gefunden wird, wird der Parameter durch sie ersetzt. Das folgende Beispiel definiert beispielsweise zwei `outputLocation`-Tags mit Dateinamen, die von Parametern abhängen.

```
<repositoryItem name= "Call_Center_Score"
  inputURI= "spsscr:///job/Call Center"
  repositoryCredentialName="localhost"
  activate="ONDEMAND"/>
  <outputLocation outputId ="spsscr:///output/output_tab_${JobParam1}.png"
    location="ChartFRAME"/>
  <outputLocation outputId="/output/output_chart_${JobParam2}.html"
    location="ReportDIV"/>
  <sourceLinkPrompt promptId="JobParam1" parameterValue="Jan" />
  <sourceLinkPrompt promptId="JobParam2"
    targetNameParameter="html_id_for_the_value" />
</repositoryItem>
```

`JobParam1` wird im Namen durch den Wert `Jan` ersetzt, wodurch `output_chart_Jan.png` bei `ChartFRAME` angezeigt wird.

Bei `JobParam2` wird der mit dem `html`-Steuerelement verknüpfte Wert für den Parameter im Namen substituiert. Falls dieser Wert `Illinois` lautet, wird bei `ReportDIV` die Datei `output_tab_Illinois.html` angezeigt.

Tag "sourceLinkPrompt"

Der Tag `sourceLinkPrompt` weist die Elementparameter den Eingabeaufforderungen zu, die ihre Werte bereitstellen. Dabei kann es sich um benutzerdefinierte HTML-Elemente, `JavaScript`-Funktionen, über den Tag `repositoryItemPrompt` erstellte Eingabeaufforderungen oder direkt angegebene Werte handeln.

Der Tag `sourceLinkPrompt` muss immer in einem Tag `repositoryItem` verschachtelt sein. Wenn das Element ausgeführt wird, werden die Parameterwerte über die `sourceLinkPrompt`-Tags abgerufen.

Tabelle 8. Attribute für den Tag "sourceLinkPrompt".

Name	Erforderlich	Beschreibung
targetNameParameter	true	Name des Parameters wie im Repository-Element definiert. Geben Sie bei Scoring-Modellen mit mehreren Tabellen den Wert als <i>Tabelle.Parameter</i> an, wobei <i>Tabelle</i> der Name der Tabelle und <i>Parameter</i> der Name des Parameters ist.
promptId	false	Die <code>promptId</code> könnte die ID eines Tags <code>reportPrompt</code> oder der Name eines HTML-Steuerelements sein. Wenn ein Eingabeaufforderungswert benötigt wird, wird der Tag <code>reportPrompt</code> oder das HTML-Steuerelement verwendet, um den Eingabeaufforderungswert zu ermitteln. <code>promptId</code> , <code>parameterValue</code> oder <code>getValueJSFunction</code> sollte angegeben werden.

Tabelle 8. Attribute für den Tag "sourceLinkPrompt" (Forts.).

Name	Erforderlich	Beschreibung
parameterValue	false	<p>Legt den Wert für den Parameter fest, anstatt ihn anzufordern. Dieser Tag sollte verwendet werden, wenn die Anwendung den Parameterwert kennt, wenn die JSP verarbeitet wird. In diesem Fall kann der Wert über dieses Attribut direkt angegeben werden.</p> <p>Wenn parameterValue angegeben ist, sollte promptId und getValueJSFunction nicht verwendet werden.</p>
getValueJSFunction	false	<p>Legt eine Funktion fest, die aufgerufen wird, um die Eingabeaufforderungswerte abzurufen. Die Funktion sollte entweder einen einzelnen Wert oder ein Wertes-Array zurückgeben.</p> <p>Dieses Attribut sollte den Funktionsnamen, Klammern und bei Bedarf auch Parameter enthalten. Für die Funktion <i>MyGetValues</i>, die einen Parameter aufnimmt, setzen Sie das Attribut beispielsweise auf <code>MyGetValues('myPromptID')</code>.</p>
validateJSFunction	false	<p>Gibt eine Funktion an, die aufgerufen wird, um eine Validierung der Eingabeaufforderung bereitzustellen. Die Funktion sollte true zurückgeben, wenn die Eingabeaufforderungen gültig sind.</p> <p>Dieses Attribut sollte den Funktionsnamen, Klammern und bei Bedarf auch Parameter enthalten. Für die Funktion <i>MyValidate</i>, die einen Parameter aufnimmt, setzen Sie das Attribut beispielsweise auf <code>MyValidate('myPromptID')</code>.</p>

Tagverschachtelung

Keine

Durchgeführte Validierungen

Keine

Erwartete Ausgabe

Keine

Verwendungsbeispiel

Folgendes Berichtsbeispiel fordert die Eingabe zweier Parameterwerte über `repositoryItemPrompt`-Tags an. Die `sourceLinkPrompt`-Tags für den Bericht *CountrySales* verwenden die IDs für diese Prompts, um ihre Werte an den Bericht weiterzugeben.

```
<r:repositoryItem name="CountrySales"
  reportDefinitionURI="spsscr:///rpts/CountryCity_cascadingParameter.rptdesign"
  repositoryCredentialName="repositoryCredential"
  outputType="HTML" activate="ONDEMAND" location="ReportDIV">
  <r:sourceLinkPrompt targetNameParameter="ShipCountry" promptId="IDFilter"/>
  <r:sourceLinkPrompt targetNameParameter="ShipCity" promptId="IDFilter1"/>
</r:repositoryItem>

<table width="95%" cellspacing="1" bgcolor="black">
  <tr bgcolor="white">
    <r:repositoryItemPrompt promptId="IDFilter"
      repositoryItemName="CountrySales1" parameterName="ShipCountry"/>
  </tr>
  <tr bgcolor="white">
    <r:repositoryItemPrompt promptId="IDFilter1"
      repositoryItemName="CountrySales1" parameterName="ShipCity"/>
  </tr>
</table>
```

Ein ähnlicher Ansatz kann bei Scoring-Modellen verfolgt werden. Folgendes Scoring-Beispiel fordert die Eingabe von fünf Parameterwerten über `input`-Tags an. Die `sourceLinkPrompt`-Tags für das Element *Configuration* verwenden die IDs für diese Eingabeaufforderungen, um ihre Werte weiterzugeben.

```
<table>
  <tr>
    <td>Age</td>
    <td><input name="Age" id="Age" type="text"/></td>
  </tr>
  <tr>
    <td>Blood Pressure</td>
    <td><input name="BP" id="BP" type="text"/></td>
  </tr>
  <tr>
    <td>Cholesterol</td>
    <td><input name="Cholesterol" id="Cholesterol" type="text"/></td>
  </tr>
  <tr>
    <td>K</td>
    <td><input name="K" id="K" type="text"/></td>
  </tr>
  <tr>
    <td>Na</td>
    <td><input name="Na" id="Na" type="text"/></td>
  </tr>
</table>

<r:repositoryItem name="MyConfiguration" inputURI="scoring:///KMeans"
  repositoryCredentialName="repositoryCredential" outputType="HTML"
  activate="ONDEMAND" location="ReportIframe">
  <r:sourceLinkPrompt targetNameParameter="Age" promptId="Age"/>
  <r:sourceLinkPrompt targetNameParameter="BP" promptId="BP"/>
  <r:sourceLinkPrompt targetNameParameter="Cholesterol"
    promptId="Cholesterol"/>
  <r:sourceLinkPrompt targetNameParameter="Drug" parameterValue="DrugX"/>
  <r:sourceLinkPrompt targetNameParameter="K" promptId="K"/>
  <r:sourceLinkPrompt targetNameParameter="Na" promptId="Na"/>
</r:repositoryItem>
```

Der Wert für den Parameter *Drug* wird auf der Seite als *DrugX* angegeben, indem das Attribut `parameterValue` verwendet wird.

Wenn die Parameter, die von der Scoring-Konfiguration verwendet werden, in verschiedenen Tabellen definiert sind, stellen Sie dem Namen des Parameters den Namen der Tabelle voran und trennen Sie die beiden Namen mit einem Punkt voneinander. Beispiel: Wenn die Parameter *Age* und *BP* in *Table1* und die restlichen Parameter in *Table2* definiert werden, werden die Elemente `sourceLinkPrompt` wie folgt angegeben:

```
<r:repositoryItem name="MyConfiguration" inputURI="scoring:///KMeans"
  repositoryCredentialName="repositoryCredential" outputType="HTML"
  activate="ONDEMAND" location="ReportIframe">
```

```

<r:sourceLinkPrompt targetNameParameter="Table1.Age" promptId="Age"/>
<r:sourceLinkPrompt targetNameParameter="Table1.BP" promptId="BP"/>
<r:sourceLinkPrompt targetNameParameter="Table2.Cholesterol"
  promptId="Cholesterol"/>
<r:sourceLinkPrompt targetNameParameter="Table2.Drug" parameterValue="DrugX"/>
<r:sourceLinkPrompt targetNameParameter="Table2.K" promptId="K"/>
<r:sourceLinkPrompt targetNameParameter="Table2.Na" promptId="Na"/>
</r:repositoryItem>

```

Tag "sourceLinkRepositoryItem"

Der Tag `sourceLinkRepositoryItem` legt das Quellelement und die Quellenvariablen fest, die verwendet werden, um die für das Element definierten Parameter zu erfüllen. Bei diesem Verfahren wird das übergeordnete Element mit den in den verschachtelten `sourceLinkVariable`-Tags definierten Parametern ausgeführt, wenn auf das Quellelement geklickt wird.

Dieser Tag muss immer in einem Tag `repositoryItem` verschachtelt sein. Er sollte mindestens einen verschachtelten Tag `sourceLinkVariable` enthalten.

Tabelle 9. Attribute für den Tag "sourceLinkRepositoryItem".

Name	Erforderlich	Beschreibung
sourceName	true	Name des <code>repositoryItem</code> , das als Quelle für die Beziehung verwendet wird
linkType	false	Legt fest, welche Aktion im Quellenbericht die Ausführung des aktuellen Berichts auslöst. Derzeit wird nur ein <code>linkType</code> , <code>row</code> , unterstützt. Für diesen Typ wird der Zielbericht ausgeführt, wenn auf eine Zeile im Quellenbericht geklickt wird. In künftigen Versionen werden gegebenenfalls weitere <code>linkTypes</code> unterstützt.

Tagverschachtelung

Der Tag `sourceLinkRepositoryItem` enthält mindestens einen Tag `sourceLinkVariable`, der die Namen der Quellenspalte und der Zielparameter angibt.

Erwartete Ausgabe

Keine

Verwendungsbeispiel

Im folgenden Beispiel wird *CityDetails* als Bericht festgelegt, der aufgrund einer Benutzeraktion im Bericht *AllCountries* ausgeführt wird.

```

<r:repositoryItem name="CityDetails"
  inputURI="spssc:///SampleReports/BIRT/CountrySalesByCity.rptdesign"
  repositoryCredentialName="repositoryCredential"
  outputType="HTML" width="400" height="300"
  activate="ONDEMAND" location="SecondReportDIV">
  <r:sourceLinkRepositoryItem sourceReportName="AllCountries">
    <r:sourceLinkVariable columnName="ShipCountry"
      targetNameParameter="ShipCountry" />
  </r:sourceLinkRepositoryItem>
</r:repositoryItem>

```

Tag "sourceLinkReport"

Dieser Tag wird nicht weiter unterstützt. Verwenden Sie stattdessen den Tag `sourceLinkRepositoryItem`.

Tag "sourceLinkVariable"

Der Tag `sourceLinkVariable` definiert die Zuordnung zwischen der Variablen oder Spalte, die im Quell-element verwendet werden soll, und dem im Zielelement definierten Parameter. Dieser Tag muss immer in einem Tag `sourceLinkRepositoryItem` verschachtelt sein.

Tabelle 10. Attribute für den Tag "sourceLinkVariable".

Name	Erforderlich	Beschreibung
columnName	true	Für Visualisierungsberichte enthält dieses Attribut die ID des Elements <code>sourceVariable</code> oder <code>derivedVariable</code> der Visualisierungsspezifikation. Derzeit werden nur kategoriale Variablen unterstützt.
targetNameParameter	true	Name des Parameters in der Zielabfrage

Tagverschachtelung

Keine

Durchgeführte Validierungen

Keine

Erwartete Ausgabe

Keine

Verwendungsbeispiel

Im folgenden Beispiel wird die Variable `ShipCountry` im Bericht `AllCountries` dem Parameter `ShipCountry` im Bericht `CityDetails` zugeordnet.

```
<r:repositoryItem name="CityDetails"
  inputURI="spsscr:///SampleReports/Vis/CitiesBarChart.viz"
  repositoryCredentialName="repositoryCredential"
  outputType="png" width="400" height="300"
  activate="ONDEMAND" location="SecondReportDIV">
  <r:sourceLinkRepositoryItem sourceName="AllCountries">
    <r:sourceLinkVariable columnName="ShipCountry"
      targetNameParameter="ShipCountry" />
  </r:sourceLinkRepositoryItem>
</r:repositoryItem>
```

Tag "actionHandler"

Definiert die Aktionshandler, die auf das Element angewendet werden sollen. Wenn Aktionshandler definiert werden, ist die automatische Verknüpfungseinstellung über `sourceLinkRepositoryItem` nicht mehr wirksam.

Der Application Builder ist für die Ausführung der Zielelemente über die öffentliche Java-Script-API `runRepositoryItem` verantwortlich.

Tabelle 11. Attribute für den Tag "actionHandler".

Name	Erforderlich	Beschreibung
event	true	Der Name des Ereignisses. Gültige Ereignisse sind: <ul style="list-style-type: none"> • <i>onclick</i> • <i>onmouseover</i> • <i>onmouseout</i>
function	true	Der Name der JavaScript-Funktion, die aufgerufen wird, wenn das Ereignis eintritt. Hier darf nur der Funktionsname ohne Klammern oder Parameter angegeben werden.
partId	false	Dies wird verwendet, um den spezifischen Teil des Berichts zu identifizieren, für den die Aktionen gelten sollen.

Tagverschachtelung

Datenwerte, die als Parameter an die JavaScript-Funktion weitergegeben werden müssen, sollten über verschachtelte Tags `actionParameter` definiert werden.

Verwendungsbeispiel

Der folgende Tag `repositoryItem` definiert drei Aktionshandler, einen für jeden Ereignistyp, der eintreten könnte. Jeder Handler ruft eine eindeutige JavaScript-Funktion auf, die die anschließende Verarbeitung definiert.

```
<r:repositoryItem name="AllCountries"
  inputURI="spsscr:///SampleReports/BIRT/CountrySales.rptdesign"
  repositoryCredentialName="repositoryCredential"
  outputType="HTML"
  width="400" height="300"
  activate="ONLOAD" location="ReportDIV">
  <r:actionHandler event="onclick" function="myOnClick">
    <r:actionParameter name="ShipCountry"/>
  </r:actionHandler>
  <r:actionHandler event="onmouseover" function="myOnOver">
    <r:actionParameter name="ShipCountry"/>
  </r:actionHandler>
  <r:actionHandler event="onmouseout" function="myOnOut" />
</r:repositoryItem>
```

Tag "actionParameter"

Für jeden Datenwert im Element, der an die JavaScript-Funktion `actionHandler` weitergegeben werden muss, sollte ein Tag `actionParameter` vorhanden sein. Dieser Tag muss immer im Tag `actionHandler` verschachtelt sein.

Tabelle 12. Attribute für den Tag "actionParameter".

Name	Erforderlich	Beschreibung
name	true	Name der Spalte oder Variable, die festlegt, welcher Wert aus den Berichtsergebnissen an die Funktion weitergegeben werden soll. Für Visualisierungsberichte ist der Name das Attribut id des Elements sourceVariable oder derivedVariable. Derzeit werden nur kategoriale Variablen unterstützt.

Tagverschachtelung

Keine

Verwendungsbeispiel

Folgendes Beispiel definiert einen actionParameter mit dem Namen *ShipCountry*, der an die JavaScript-Funktion myOnClick weitergeben wird, wenn der Benutzer auf den Bericht klickt.

```
<r:actionHandler event="onclick" function="myOnClick">
  <r:actionParameter name="ShipCountry"/>
</r:actionHandler>
```

Tagbibliothekbeans

Das Framework enthält Tagbibliothekbeans, die zusammen für verschiedene Zwecke verwendet werden können. Sie können beispielsweise zum Abrufen eines Datensets verwendet werden, das dann verwendet werden kann, um benutzerdefinierte HTML-Steuerelemente zu erstellen.

Um die Beans zu verwenden, müssen Sie zunächst auf der JSP Verweise auf sie erstellen. Das geschieht über das Attribut import der page-Aufbauregel.

```
<%@ page contentType="text/html; charset=utf-8"
  language="java"
  session="true"
  import="java.util.Map"
  import="java.util.HashMap"
  import="com.spss.report.taglib.bean.ReportBean"
  import="com.spss.report.taglib.bean.Credential"
%>
```

Diese Codebeispiele für Beans verwenden die Tagbibliothek JSTL (JavaServer Pages Standard Tag Library), die über die taglib-Aufbauregel integriert werden sollte.

```
<%@ taglib uri="http://java.sun.com/jstl/core" prefix="c" %>
```

Weitere Informationen zu JSTL finden Sie in der Sun-Dokumentation.

Credential-Bean

Die Credential-Bean definiert die Berechtigungsnachweise, die von anderen Beans verwendet werden.

Das folgende Codebeispiel erstellt zwei Berechtigungsnachweise und speichert sie in einer Hashzuordnung (HashMap). Im folgenden Beispiel stellt der Berechtigungsnachweis *localhost* die Anmeldeinformationen für IBM SPSS Collaboration and Deployment Services Repository bereit. Die Berechtigungsnachweise *ps4008* gelten für die SQL Server-Datenquelle "ps4008", die von der Berichtsdefinition referenziert wird.

```
<%
Map credentialMap = new HashMap();
Credential repositoryCredential = new Credential("localhost","Native","admin","spss",null);
```

```
Credential datasourceCredential = new Credential("ps4008",null,"sa","sa",null);
credentialMap.put("localhost",repositoryCredential);
credentialMap.put("ps4008",datasourceCredential);
%>
```

ReportBean-Bean

Die ReportBean-Bean wird verwendet, um die Daten für ein Dataset abzurufen, das in einer Berichtsdefinition definiert ist.

Der folgende Code verwendet die zuvor erstellte credentialMap, um ein Dataset abzurufen. Visualisierungsberichte unterstützen diese Funktion nicht.

```
<%-- Creating JavaBeans --%>
<jsp:useBean id="report" class="com.spss.report.taglib.bean.ReportBean">
  <jsp:setProperty name="report" property="reportDefinitionURI"
    value="file:///d:/SPSS/ps4008/Test.dbq" />
  <jsp:setProperty name="report" property="repositoryCredentialName"
    value="localhost" />
  <jsp:setProperty name="report" property="host" value="localhost" />
  <jsp:setProperty name="report" property="port" value="8080" />
  <jsp:setProperty name="report" property="dataSetName"
    value="DataSet1" />
  <jsp:setProperty name="report" property="credentialMap"
    value="<%=credentialMap%>" />
</jsp:useBean>
```

Folgende Eigenschaften werden in diesem Code verwendet:

- Die Eigenschaft *reportDefinitionURI*, die den Speicherort des Berichts angibt.
- Die Eigenschaft *repositoryCredentialName*, die den Host angibt.
- Die Eigenschaft *port*, die den Port definiert.
- Die Eigenschaft *dataSetName*, die den Namen des Datasets angibt, wie er in der Berichtsdefinition für Berichte von BIRT Report Designer for IBM SPSS definiert ist.
- Die Eigenschaft *credentialMap*, die einen Verweis auf eine HashMap mit den zu verwendenden Berechtigungenachweisen definiert.

Die ReportBean-Bean kann anschließend ausgeführt werden, um das Dataset zurückzugeben. Die Daten können verwendet werden, um wie im folgenden Code dargestellt, ein Listenfeld zu erstellen.

```
<SELECT style="WIDTH :250 px" ID="EmployeeID_Prompt" NAME="EmployeeID_Prompt"
  TABINDEX="2">
<c:forEach var="row" items="{report.rows}">
  <c:forEach var="column" items="{row.columns}">
    <c:if test='${column.name == "EmployeeID"}'>
      <OPTION VALUE='<c:out value="{column.value}" />'>
        <c:out value="{column.value}" />
      </OPTION>
    </c:if>
  </c:forEach>
</c:forEach>
</SELECT>
```

SearchBean-Bean

Die SearchBean-Bean bietet einen Abfragemechanismus für das Auffinden von Inhalt im Repository, der bestimmten Kriterien entspricht.

Die Bean kann zum Beispiel eine Liste mit Datenproviderdefinitionen und Datendateiquellen von IBM SPSS Statistics (.sav) in IBM SPSS Collaboration and Deployment Services Repository abrufen, die einem bestimmten Suchkriterium entsprechen. Der folgende Code definiert Bean-Eigenschaften zum Abfragen aller Datenproviderdefinitionen und Datenquellen von IBM SPSS Statistics, die die MIME-Typen verwenden, die diesen Quellen zugeordnet sind.

```
<jsp:useBean id="data_sources"
  class="com.spss.report.taglib.bean.SearchBean" scope="page">
  <jsp:setProperty name="data_sources" property="request"
    value="<%= request %>" />
  <jsp:setProperty name="data_sources" property="credentialName"
```

```

    value="AuthenticationCredential" />
<jsp:setProperty name="data_sources" property="searchQuery"
    value="<%= "('$search/mimetype'='application/x-vnd.spss-spss-data' or "
    + "'$search/mimetype'='application/x-vnd.spss-statistics-data' or "
    + "'$search/mimetype'='application/x-vnd.spss-data-provider')" %>" />
</jsp:useBean>

```

Folgende Eigenschaften werden in diesem Code verwendet:

- Die Eigenschaft *request*, die ein `HttpServletRequest`-Objekt definiert.
- Die Eigenschaft *credentialName*, die die für eine Verbindung zum Repository erforderlichen Berechtigungsnachweise angibt. In diesem Fall entspricht der Wert dem Berechtigungsnachweis *AuthenticationCredential*, der mithilfe des Tags `credential` definiert wurde.
- Die Eigenschaft *searchQuery*, die eine Zeichenfolge angibt, die das Suchkriterium enthält.

Die `SearchBean`-Bean kann anschließend ausgeführt werden, um die übereinstimmenden Datenquellen zurückzugeben. Der folgende Code gibt Name, Änderungsdatum, Versionsbeschriftung und Autormetadaten für die Datenquellen in einer Tabelle an.

```

<Table border="0" height="100%" width="100%" cellpadding="0" cellspacing="0">
  <tr>
    <td align="center" bgcolor="#EEEEEE">
      Datenquelle
    </td>
    <td align="center" bgcolor="#EEEEEE">
      Änderungsdatum
    </td>
    <td align="center" bgcolor="#EEEEEE">
      Versionsbeschriftung
    </td>
    <td align="center" bgcolor="#EEEEEE">
      Autor
    </td>
  </tr>
  <c:forEach var="data_source" items="${data_sources.records}"
    varStatus="status" begin="0" end="3" step="1">
    <tr>
      <td align="center" bgcolor="#EEEEEE">
        <c:out value="${data_source.title}" />
      </td>
      <td align="center" bgcolor="#EEEEEE">
        <c:out value="${data_source.modifiedDate}" />
      </td>
      <td align="center" bgcolor="#EEEEEE">
        <c:out value="${data_source.versionLabel}" />
      </td>
      <td align="center" bgcolor="#EEEEEE">
        <c:out value="${data_source.author}" />
      </td>
    </tr>
  </c:forEach>
</Table>

```

PevMetaDataBean-Bean

Die `PevMetaDataBean`-Bean ruft Variablenmetadaten aus Datenproviderdefinitionen und Datendateiquellen von IBM SPSS Statistics (.sav) ab.

Der folgende Code definiert Eigenschaften für die Bean zum Abfragen einer .sav-Datei.

```

<jsp:useBean id="variables"
    class="com.spss.report.taglib.bean.PevMetaDataBean" scope="page">
  <jsp:setProperty name="variables" property="request" value="<%= request %>" />
  <jsp:setProperty name="variables" property="dataseturi"
    value="spsscr:///sav_files/demo.sav" />
  <jsp:setProperty name="variables" property="credentialName"
    value="AuthenticationCredential" />
</jsp:useBean>

```

Folgende Eigenschaften werden in diesem Code verwendet:

- Die Eigenschaft *request*, die ein `HttpServletRequest`-Objekt definiert.
- Die Eigenschaft *dataseturi*, die den URI für die Datendatei oder die Datenproviderdefinition angibt, die die Variablen enthält.

- Die Eigenschaft *credentialName*, die die für eine Verbindung zum Repository erforderlichen Berechtigungsnachweise definiert. In diesem Fall entspricht der Wert dem Berechtigungsnachweis *AuthenticationCredential*, der mithilfe des Tags `credential` definiert wurde.

Die `PevMetadataBean`-Bean kann anschließend ausgeführt werden, um die Metadaten für die Variablen im Dataset zurückzugeben. Der folgende Code stellt die Metadaten in einer Tabelle dar.

```
<Table border="0" height="100%" width="100%" cellpadding="0" cellspacing="0">
  <tr>
    <td align="center" bgcolor="#EEEEEE">
      Variablenname
    </td>
  </tr>
  <tr>
    <td align="center" bgcolor="#EEEEEE">
      <c:forEach var="group" items="{variables.variablesMetaData}" >
        <c:forEach var="v" items="{group.variableMetaData}" varStatus="status"
          begin="0" end="3" step="1">
          <tr>
            <td align="center" bgcolor="#EEEEEE">
              <c:out value="{status.count}" /> <c:out value="{v.name}" />
            </td>
          </tr>
        </c:forEach>
      </c:forEach>
    </td>
  </tr>
</Table>
```

ScoringBean-Bean

Die `ScoringBean`-Bean ruft eine Liste von Scoring-Konfigurationen für ein angegebenes Modell ab, die auf eine Scoring-Anforderung reagieren können.

Die Methode `getScoringConfigurations` der Bean akzeptiert die folgenden Parameter:

- Den Parameter *credential*, der die Berechtigungsnachweise für den Zugriff auf IBM SPSS Collaboration and Deployment Services Repository angibt, die mit der `Credential`-Bean definiert wurden.
- Den Parameter *modelLocationUri*, der den URI für ein Modell in IBM SPSS Collaboration and Deployment Services Repository angibt.

Statt einem `Credential`-Bean-Objekt können die folgenden beiden Parameter zur Angabe von Berechtigungsnachweisen verwendet werden:

- Den Parameter *request*, der ein `HttpServletRequest`-Objekt angibt.
- Den Parameter *credentialName*, der die Berechtigungsnachweise zur Herstellung einer Verbindung zu IBM SPSS Collaboration and Deployment Services Repository angibt, die mit dem Tag `credential` definiert wurden.

Der folgende Code ruft für das Modell *KMeans.xml* die Scoring-Information ab, die auf eine Scoring-Anforderung mit einem durch den Tag `credential` definierten Berechtigungsnachweis antworten kann.

```
<r:credential name="repositoryCredential" provider="Native"
  username='<%= request.getParameter("userid")%>'
  password='<%= request.getParameter("password")%>' />

<%
  String[] configurations = ScoringBean.getScoringConfigurations(request,
    "repositoryCredential", "spsscr:///Sample/KMeans.xml");
%>
```

Das von der Bean zurückgegebene Array kann verwendet werden, um ein Formular zu füllen, aus dem ein Benutzer eine für künftige Scoring-Vorgänge zu verwendende Scoring-Konfiguration auswählen kann.

```
<form id="selectConfigurationForm" target="ScoringIframe" method="POST">
  <div style="display:none">
    <input name="userid" type="text"
      value="<%= request.getParameter("userid")%>"/>
    <input name="password" type="text"
      value="<%= request.getParameter("password")%>"/>
  </div>
  Scoring-Konfiguration auswählen:
  <select name="selectedConfiguration"
    onchange="onSelectConfiguration(this)">
    <option></option>
  </select>
```

```

        for (int i=0; i < configurations.length; i++)
        {
    %>
            <option value="<%= configurations[i].replaceAll(" ", "%20")%>"
                <%= configurations[i] %></option>
    <%
        }
    %>
    </select>
</form>

```

JSP-Beispiele

IBM SPSS Collaboration and Deployment Services enthält eine Vielzahl an JSP-Beispielen, die die Verwendung der Tagbibliothek illustrieren.

Die Beispiele sind in folgenden Kategorien zusammengefasst:

- **Berichterstellung.** Interaktive Nutzung von Berichten von BIRT Report Designer for IBM SPSS und Visualisierungsberichten einschließlich der Ausführung eines zweiten Berichts aufgrund einer Auswahl. Über folgenden Link können Sie auf die Beispiele zugreifen:

`http://<Servername>:<Port>/IBMSPPSTagLib/index.html`

- **Scoring.** Generieren von Scores für ein Vorhersagemodell, das zum Scoring konfiguriert wurde. Schließt mehrere Ansätze zum Bereitstellen konfigurierter Modelle mit Daten zum Scoring ein. Über folgenden Link können Sie auf die Beispiele zugreifen:

`http://<Servername>:<port>/scoringTagLib/index.html`

Wenn über die URL für ein Beispielset keine Einführungsseite aufgerufen wird, wurde die WAR-Datei mit den Beispielen nicht auf dem Server für IBM SPSS Collaboration and Deployment Services Repository bereitgestellt. Die bereitzustellenden WAR-Dateien und erforderlichen Beispieldateien befinden sich im Verzeichnis `./components/taglib/Samples/TagLib` der Repository-Installation. Stellen Sie die WAR-Datei gemäß der Dokumentation für Ihren Anwendungsserver bereit.

Anmerkung: Um Konflikte mit anderen Anwendungen zu vermeiden, die auf Ihrem Server ausgeführt werden, können Sie für die bereitgestellten Beispiele ein benutzerdefiniertes Kontextstammverzeichnis angeben. Weitere Informationen finden Sie in der Hilfe zu Ihrem Anwendungsserver.

Klicken Sie auf der Einführungsseite für die Beispiele für ein beliebiges Beispiel auf **Quelle anzeigen**, um dessen Quellcode anzuzeigen. Um die Funktionalität kennenzulernen, führen Sie die Beispiele über die Seite aus, indem Sie auf **Ausführen** klicken. Um eine erfolgreiche Ausführung zu ermöglichen, sind jedoch folgende Elemente erforderlich:

- Beispielressourcen in einer bestimmten Ordnerstruktur in IBM SPSS Collaboration and Deployment Services Repository.
- Gültige Berechtigungsnachweise für den Zugriff auf die Ressourcen, auf die die Beispiele verweisen.

Anweisungen zur Konfiguration der Umgebung für erfolgreiche Beispielausführung stehen auf der Einführungsseite für die Beispiele zur Verfügung.

Kapitel 4. Portalintegration

Die Web-Service-Architektur von IBM SPSS Collaboration and Deployment Services bietet die Möglichkeit der Integration mit Portalservern. Dadurch können stark angepasste Inhalte über Plug-in-Komponenten der Benutzerschnittstellenkomponente bereitgestellt werden, die über Web-Services Markupcode-Fragmente produzieren, die auf einer Portalseite zusammengestellt werden.

In der Regel wird eine Portalseite als eine Gruppe von nicht überlappenden Fenstern angezeigt, wobei jedes Fenster ein Inhaltssegment enthält. Beispiele für Portalanwendungen sind E-Mails, Wetterberichte, Diskussionsforen und Nachrichten. Portale für IBM SPSS Collaboration and Deployment Services können ebenfalls verwendet werden, um angepasste Inhalte wie Ausgaben von Berichten und analytischen Verarbeitungen, Tabellen, Diagrammen usw. bereitzustellen.

Das Repository unterstützt Portalintegration basierend auf dem Standard JSR 168. Der Standard JSR 168, der von der Java Community Process Group (<http://jcp.org>) vorgeschlagen wurde, ermöglicht die Interoperabilität für Portlets zwischen unterschiedlichen Webportalen. Diese Spezifikation definiert eine Reihe von APIs für die Zusammenarbeit zwischen dem Portlet-Container und dem Portlet und befasst sich mit Bereichen wie Personalisierung, Präsentation und Sicherheit. Die Implementierung von JSR 168 umfasst IBM Web Portal von WebSphere, Oracle Application Server Portal 10g, BEA WebLogic Portal, Vignette Portal, Sun Portal Server und JBoss.

Das Repository unterstützt zudem die Portalintegration mit Microsoft SharePoint Server und Web Parts.

Offiziell unterstützte Portalumgebungen:

- WebSphere Portal Server 6.1
- WebLogic Portal Server 10.0
- GateIn 3.5.0 (JBoss AS7)
- Sun Java Enterprise System 5
- Microsoft Sharepoint 2010 Server
- Microsoft Sharepoint 2007 Server

Das Repository kann auch mit anderen Portalumgebungen auf Basis von JSR 168 und J2SE 5.0 integriert werden.

IBM SPSS Collaboration and Deployment Services Portlet und IBM SPSS Collaboration and Deployment Services Web Part können verwendet werden, um Repository-Inhalte für Portalbenutzer bereitzustellen. Die Architektur ermöglicht zudem die Erstellung von benutzerdefinierten JSR 168-kompatiblen Share-Point Web Parts, die Web-Services von IBM SPSS Collaboration and Deployment Services verwenden.

Einschränkung: Ihr Browser muss Cookies unterstützen, damit Sie Portalintegration für den Zugriff auf Repository-Inhalte verwenden können.

Installation

Portalkomponenten von IBM SPSS Collaboration and Deployment Services werden auf Installationsdatenträger 2 in `/PORTLET` als `IBMSPSSPortlet.war` (Portlet) und `IBMSPSSWebPart.wsp` (Web Part) bereitgestellt.

Installation von IBM SPSS Collaboration and Deployment Services Portlet

Die Installation von `IBMSPSSPortlet.war` hängt vom jeweiligen Portalservertyp ab. Genauere Informationen finden Sie in der Dokumentation des Portalserveranbieters.

Installation von IBM SPSS Collaboration and Deployment Services Web Part

Installationsanforderungen für SharePoint Web Part:

- Microsoft SharePoint 2007
- Microsoft Web Service Enhancement 2.0 (WSE 2.0 SP3)

So installieren Sie IBM SPSS Collaboration and Deployment Services Web Part:

1. Kopieren Sie die Datei *IBMSPSSWebPart.wsp* vom Repository-Installationsdatenträger an eine vordefinierte Position auf dem SharePoint-Host, beispielsweise in *c:\temp*.
2. Führen Sie über das Verzeichnis */bin* der SharePoint-Serverinstallation folgende Befehle aus:

```
stsadm -o addsolution -filename c:\tmp\IBMSPSSwebpart.wsp  
stsadm -o deploysolution -name IBMSPSSwebpart.wsp -immediate  
-allowgacdeployment -url http://<Hostname>
```
3. Verwenden Sie die Verwaltungsdienstprogramme von SharePoint, um Web Part der Web Part-Galerie hinzuzufügen und es anschließend bereitzustellen. Weitere Informationen finden Sie in der Dokumentation zu Microsoft SharePoint.

Nach der Installation muss die Komponente für den Zugriff auf eine bestimmte Ressource im Repository konfiguriert werden. Zudem müssen Vorgaben für die Komponente eingerichtet werden.

Konfiguration

Nach der Installation der Portalkomponente und der Fertigstellung des Seitenlayouts für das Portal werden Sie aufgefordert, die Komponente für den Zugriff auf eine Repository-Ressource zu konfigurieren. Im Allgemeinen werden bei der Konfiguration des Portalzugriffs der Repository-Server definiert, die Repository-Berechtigungs-nachweise angegeben, die Ressource, die im Portal bereitgestellt wird, wird ausgewählt und bei Bedarf werden Berechtigungs-nachweise für Datenquellen und Standard-Eingabeaufforderungswerte festgelegt. Zudem können Sie das Erscheinungsbild und das Verhalten der Komponenten konfigurieren, indem Sie die entsprechenden Vorgaben vornehmen.

Konfigurieren von IBM SPSS Collaboration and Deployment Services Portlet

Öffnen Sie die Portletkonfigurationsseite. Die Seite wird je nach Portalservertyp unterschiedlich geöffnet.

1. Geben Sie die URL des Repository-Servers an.
2. Geben Sie die Benutzerberechtigungs-nachweise für das Repository und den Sicherheitsprovider für die Anmeldeauthentifizierung an.
3. Wählen Sie die Repository-Ressource aus, die im Portal bereitgestellt werden soll. Stellen Sie sicher, dass die richtige Ressourcenversion angegeben ist.
4. Geben Sie bei Bedarf die Berechtigungs-nachweise für die Datenquelle an, die von der Ressource referenziert wird. Wenn ein Bericht beispielsweise eine Datenbank verwendet, müssen die Berechtigungs-nachweise für die Datenbank angegeben werden. Beachten Sie, dass je nach Ressource gegebenenfalls die Berechtigungs-nachweise für mehrere Datenquellen angegeben werden müssen.
5. Wenn die Ressource Eingabeaufforderungen enthält (zum Beispiel bietet einen Bericht unter Umständen die Möglichkeit der dynamischen Auswahl von Werten), geben Sie die Standard-Eingabeaufforderungseinstellungen ein.
6. Prüfen Sie, ob die Konfigurationsinformationen korrekt sind. Um von Neuem zu beginnen, klicken Sie auf **Aktualisieren**.
7. Klicken Sie auf **Weiter**, um mit der Anzeige der Ressource fortzufahren.

Portleteinstellungen können nach der Erstkonfiguration bearbeitet werden. Zum Beispiel kann auf eine andere Repository-Ressource verwiesen werden, falls erforderlich.

Bestimmte Aspekte des Erscheinungsbilds und des Verhaltens des Portlets werden über seine Vorgaben festgelegt. Folgende Vorgaben stehen zur Verfügung:

Vorgabe	Beschreibung
<i>expiration-cache</i>	Ablaufzeitraum für den Portlet-Cache, d. h. die Zeit in Sekunden, nach der die Portletausgabe abläuft. -1 definiert, dass die Ausgabe nie abläuft. Der Standardwert ist 600.
<i>log-messages</i>	Gibt an, ob Portletnachrichten an die Protokolldatei für den Portalserver angehängt werden. Der Standardwert ist NO.
<i>reenter-dsLogin</i>	Gibt an, ob der Benutzer die Berechtigungsnachweise für die Datenquelle für die Portletinstanz bei jeder Anmeldung beim Portal angeben muss. Der Standardwert ist NO.
<i>reenter-parameter</i>	Gibt an, ob der Benutzer die Eingabeaufforderungswerte für die Portletinstanz bei jeder Anmeldung beim Portal erneut eingeben muss. Der Standardwert ist NO.
<i>refresh-parameter</i>	Legt fest, ob der Benutzer unterschiedliche Parameterwerte eingeben und den Inhalt basierend auf diesen Werten erneut anzeigen kann. Der Standardwert ist NO.
<i>use-single-sign-on</i>	Gibt an, ob das Portlet mit Single Sign-on verwendet wird. Der Standardwert ist NO.
<i>validate-input-parameter</i>	Aktiviert die Validierung von Eingabeparametern, um vor site-übergreifenden Scripting-Attacken zu schützen. Der Standardwert ist YES.
<i>window-height</i>	Die Höhe des Portletfensters (in Pixel). Der Standardwert ist 750.
<i>window-title</i>	Beschreibender Name für die Portletinstanz.
<i>window-width</i>	Die Breite des Portletfensters (in Prozent). Der Standardwert ist 100 %.

Vorgaben werden über die Serveradministrationsdienstprogramme festgelegt. Der Zugriff auf diese Programme ist vom Servertyp abhängig.

Konfigurieren von IBM SPSS Collaboration and Deployment Services Web Part

Die Webkomponentenkonfiguration umfasst dieselben grundlegenden Schritte wie die Portletkonfiguration: Konfiguration des Zugriffs auf die Repository-Ressource und die Konfigurationsoption. Beachten Sie, dass die Anzahl der angezeigten Elemente im Repository-Baum (bei der Auswahl der Ressource) über eine zusätzliche Konfigurationsoption gesteuert wird.

Single Sign-on

IBM SPSS Collaboration and Deployment Services ermöglicht Zugang über Single Sign-on, wobei eine spezielle Konfiguration des Portalserver erforderlich sein kann, um ihn für das Portlet oder die Webkomponente zu aktivieren. Die Prozeduren für die Aktivierung des Single Sign-on unterscheiden sich je nach Portalserver. Weitere Informationen finden Sie in der Dokumentation zu Ihrem Portalserver.

Bekannte Probleme

- Wenn das Portlet mit dem JBoss-Portal verwendet wird, kann die Repository-Baumansicht möglicherweise nicht erweitert werden. Um dieses Problem zu beheben, ändern Sie in der Datei `<JBoss-Installationsordner>/bin/run.bat` (`run.sh` unter UNIX) die Neugenerierung und die permanente Generierungsgröße, indem Sie `JAVA_OPTS` die folgenden Argumente hinzufügen:

```
-XX:MaxNewSize=256m -XX:MaxPermSize=256m
```
- Cookie-Einstellungen im Safari-Browser können verhindern, dass einige Repository-Artefakte im Portlet angezeigt werden, ohne dass zuvor Berechtigungsnachweise angefordert wurden. Die Cookie-Einstellung des Browsers sollte auf *Immer* anstelle von *Nur von Websites, die ich besuche* eingestellt sein, um wiederholte Anforderungen von Berechtigungsnachweisen zu verhindern.

Kapitel 5. HTML-Archiv

Ein HTML-Bericht beinhaltet normalerweise einige HTML-Dateien, die eine Reihe von referenzierten Bildern anzeigen, wobei Style-Sheets verwendet werden, um das Erscheinungsbild der Ausgabe zu kontrollieren. Aufgrund der Anzahl der enthaltenen Dateien können die Verwaltung und die gemeinsame Verwendung dieser Ausgabe eine Herausforderung darstellen. Falls eine Datei fehlt oder falsch referenziert wird, werden die Seiten nicht korrekt dargestellt.

Das HTMLC-Format (HTML Archive) adressiert das Problem der Verwaltung zahlreicher intern verknüpfter Dateien, indem es alle zugeordneten HTML-Artefakte in einer einzelnen, browserübergreifenden Datei zusammenfasst. IBM SPSS Collaboration and Deployment Services Repository enthält einen Viewer, der die Anzeige des Archivinhalts über eine Reihe von Clientanwendungen ermöglicht. Beim Zugriff auf eine im Repository gespeicherte HTMLC-Datei werden Querverweise innerhalb des Archivs im Hintergrund durch vollständige Pfade ersetzt, welche die Archivdatei referenzieren. Dies ermöglicht die vollständige Auflösung von Links zu Dateien im Archiv, sodass diese korrekt dargestellt werden.

Dateistruktur

Eine HTMLC-Archivdatei enthält folgende Elemente:

- Eine primäre HTML-Datei im Stammverzeichnis des Archivs. Beim Rendern eines HTMLC-Archivs verwendet der Viewer die erste Datei mit einer *.html*-Erweiterung im Stammverzeichnis des Archivs als die primäre Datei.
- Sekundäre Dateien, die von der primären Datei referenziert werden, z. B. Cascading Style-Sheets, Bilder, Javascript oder andere HTML-Dateien. Sekundäre Dateien können sich in jedem Ordner innerhalb des Archivs befinden.

Für alle Referenzen auf Dateien innerhalb des Archivs sollten relative Pfade verwendet werden.

Erstellen von HTMLC-Dateien

HTMLC-Dateien können bei der Arbeit mit Berichtsdesigns, die in IBM SPSS Collaboration and Deployment Services Repository gespeichert sind, in BIRT Report Designer for IBM SPSS erstellt werden. Benutzerdefinierte HTMLC-Dateien können jedoch auch über ein Programm zur Dateiarchivierung, z. B. über das Java-Archivierungstool oder WinZip, erstellt werden. Manuelles Erstellen einer HTMLC-Datei:

1. Erstellen Sie im Dateisystem die Struktur für die Dateien.
2. Erstellen Sie ein Archiv, das diese Dateien und Ordner enthält, und geben Sie dabei die Dateierweiterung *.htmlc* für die Ausgabedatei an.

Die Dateien im Archiv können manuell oder automatisch erstellt werden. In IBM SPSS Statistics können Sie die Ergebnisse einer Analyse z. B. als HTML exportieren. Die resultierenden HTML- und Bilddateien können als HTMLC-Datei archiviert werden. Alternativ können Sie einen HTML-Editor verwenden, um zu archivierende Seiten manuell zu erstellen.

Beispiel für eine benutzerdefinierte HTMLC-Datei

In diesem Beispiel wird ein Ordner dargestellt, der die Datei *gss.html* und die Unterordner *css* und *images* enthält. Die HTML-Datei referenziert Bilder im Ordner *images* und verwendet Stile aus einem Cascading Style Sheet im Ordner *css*. Unter Verwendung des Java-Archivierungstools erstellt der folgende Befehl eine HTMLC-Datei namens *custom.HTMLC*, welche die Dateien enthält.

```
jar -cvfM custom.HTMLC gss.htm images css
```

Durch Speichern dieses einzelnen Archivs im Repository kann die `gss.html`-Seite in Repository-Clients, wie IBM SPSS Collaboration and Deployment Services Deployment Portal oder IBM SPSS Collaboration and Deployment Services Deployment Manager, angezeigt werden, wobei ihre referenzierten Grafiken die definierten Stile verwenden.

Kapitel 6. Anpassungsbeispiel

Die Modellverwaltungsseite von IBM SPSS Collaboration and Deployment Services Deployment Portal bietet die Möglichkeit zur kontinuierlichen Überwachung der Leistung der in IBM SPSS Collaboration and Deployment Services Repository implementierten Modelle. Diese Modelldateien sind mit Jobs verknüpft, deren Ausführung bei Bedarf angefordert oder geplant werden kann. Die Dateien werden durch Verwendung von IBM SPSS Modeler erstellt. Modellevaluierungs- und Champion-Challenger-Jobs werden mithilfe von IBM SPSS Collaboration and Deployment Services Deployment Manager eingerichtet und ausgeführt, IBM SPSS Collaboration and Deployment Services Deployment Portal wird nur zum Anzeigen der Ergebnisse verwendet. Die als Bereiche auf der Seite "Modellverwaltung" angezeigten Informationen können folgende Elemente enthalten:

- Liste der Modelle mit der besten und schlechtesten Leistung
- Trends der Modelleistung
- Champion-Modelle
- Liste aller verfügbaren Modelldateien

Die Optionen im Konfigurationsbereich ermöglichen die Definition der Anzeigeparameter sowie das Ein- bzw. Ausblenden einzelner Registerkarten.

Informationen zur Verwendung der Seite "Modellverwaltung" finden Sie in der Hilfe zu IBM SPSS Collaboration and Deployment Services Deployment Portal.

Die Benutzerschnittstelle besteht im Wesentlichen aus einer einzelnen JavaServer Page (JSP), der *MMD-Master.jsp*. Bei den Komponenten der Benutzerschnittstelle, die auf der Seite gerendert werden, handelt es sich entweder um Berichte von BIRT Report Designer for IBM SPSS oder um Visualisierungsberichte. Diese Berichte werden mit IBM SPSS Collaboration and Deployment Services Tag Library gerendert. Die Seite selbst ist über das Framework "Tab Extension" in IBM SPSS Collaboration and Deployment Services Deployment Portal integriert.

IBM SPSS Collaboration and Deployment Services Tag Library

IBM SPSS Collaboration and Deployment Services Tag Library unterstützt die Ausführung der Berichte von BIRT Report Designer for IBM SPSS und der Visualisierungsberichte, die einen Großteil des Inhalts der Seite "Modellverwaltung" erzeugen.

Die Tagbibliothek unterstützt zudem die Interaktivität zwischen Berichten, sodass ein Quellenbericht einen Zielbericht aufrufen kann. Der Quellenbericht gibt Parameter an den Zielbericht zur Verarbeitung weiter.

Berichtsdefinitionen

Die Berichtsdefinitionen, die von der Seite "Modellverwaltung" verwendet werden, sind in folgendem Verzeichnis in der Installation von IBM SPSS Collaboration and Deployment Services Repository gespeichert:

```
<Installationsverzeichnis>/components/peb-mmd/reports
```

Um die Berichte zu untersuchen, öffnen Sie die Dateien in BIRT Report Designer for IBM SPSS. Die Visualisierungsberichte können über die IBM SPSS Visualization Designer oder einen Text- oder XML-Editor geöffnet werden.

Die Berichte werden für Referenzzwecke bereitgestellt und sollten nicht direkt bearbeitet werden. Änderungen an den Berichten werden von IBM nicht unterstützt. Sie können die Berichte jedoch kopieren und die Kopien je nach Bedarf ändern.

Ausführen von Berichten von BIRT Report Designer for IBM SPSS

Auf der Seite "Modellverwaltung" werden vier Masterberichte auf vier Registerkarten bereitgestellt. Jede Registerkarte entspricht einem Masterbericht von BIRT Report Designer for IBM SPSS.

Wenn die Registerkarte geladen wird, wird der Masterbericht, der dieser Registerkarte entspricht, über das Framework von IBM SPSS Collaboration and Deployment Services Tag Library ausgeführt.

Für jeden Bericht ist auf der Master-JavaServer Page ein Tag vorhanden. Das vorliegende Beispiel zeigt den Tag, der für die Registerkarte "Leistung" verwendet wird.

```
<!-- Tag, der den Bericht darstellt -->
<ibmspss-taglib:repositoryItem
  name="Performance_Tab_Report_Tag"
  inputURI="<<Pfad des Leistungsberichts>>"
  repositoryCredentialName="localhost"
  activate="ONLOAD"
  location="Performance_Tab_Report_Output"
  outputType="HTML">
  <ibmspss-taglib:sourceLinkPrompt
    targetNameParameter="LeastPerformingScenarios"
    parameterValue="<<localized text>>" />

<!--
  <<<< Sonstige Parameter, die lokalisierten Text darstellen>>>>
-->

<!-- Dieser Wert kommt aus benutzerdefinierten Eingabeaufforderungen -->
<ibmspss-taglib:sourceLinkPrompt
  promptId="Performance_Tab_NumberOfPerformers_Prompt"
  targetNameParameter="NumberOfPerformers"/>
<!-- Der Wert dieses Parameters wird direkt im Tag angegeben -->
<ibmspss-taglib:sourceLinkPrompt
  targetNameParameter="RunsFromDate"
  parameterValue="<%=scenariosFrom%>"/>
<ibmspss-taglib:sourceLinkPrompt
  targetNameParameter="RunsToDate"
  parameterValue="<%=scenariosTo%>"/>
</ibmspss-taglib:repositoryItem>
```

Tag "repositoryItem"

Die Berichtsinformationen werden im Tag repositoryItem angegeben.

```
<!-- Tag, der den Bericht darstellt -->
<ibmspss-taglib:repositoryItem
  name="Performance_Tab_Report_Tag"
  inputURI="<<Pfad des Leistungsberichts>>"
  repositoryCredentialName="localhost"
  activate="ONLOAD"
  location="Performance_Tab_Report_Output"
  outputType="HTML">
  <ibmspss-taglib:sourceLinkPrompt
    targetNameParameter="LeastPerformingScenarios"
    parameterValue="<<localized text>>" />

<!--
  <<<< Sonstige Parameter, die lokalisierten Text darstellen>>>>
-->

<!-- Dieser Wert kommt aus benutzerdefinierten Eingabeaufforderungen -->
<ibmspss-taglib:sourceLinkPrompt
  promptId="Performance_Tab_NumberOfPerformers_Prompt"
  targetNameParameter="NumberOfPerformers"/>
<!-- Der Wert dieses Parameters wird direkt im Tag angegeben -->
<ibmspss-taglib:sourceLinkPrompt
  targetNameParameter="RunsFromDate"
  parameterValue="<%=scenariosFrom%>"/>
<ibmspss-taglib:sourceLinkPrompt
  targetNameParameter="RunsToDate"
  parameterValue="<%=scenariosTo%>"/>
</ibmspss-taglib:repositoryItem>
```

Der Tag `repositoryItem` hat folgende Attribute:

- Das Attribut *name*, das den Tag "repositoryItem" eindeutig kennzeichnet. Die öffentliche API `runRepositoryItem` verwendet diesen Namen, um den Bericht darzustellen. Für die Registerkarte "Leistung" ist der Name *Performance_Tab_Report_Tag*.
- Das Attribut *inputURI*, das den Speicherort des Berichts angibt. Für die Seite "Modellverwaltung" werden alle Berichte vom Dateisystem des Servers im Verzeichnis `peb-mmd` der Installation von IBM SPSS Collaboration and Deployment Services Repository abgerufen. Der angegebene URI muss gültig sein.
- Das Attribut *activate* legt fest, wann der Bericht ausgeführt wird. Für die Registerkarte "Leistung" ist der Wert *ONLOAD*, d. h., der Bericht wird ausgeführt, wenn die Seite geladen wird. Der Wert *ONDEMAND* gibt an, dass der Benutzer dafür verantwortlich ist, die Ausführung des Berichts zu starten, indem er die öffentliche API `runReport` aufruft, die in der Tagbibliothek "Reporting" bereitgestellt wird. Weitere Informationen finden Sie im Thema „JavaScript-API“ auf Seite 61.
- Das Attribut *location*, das den Speicherort angibt, an dem der Bericht gerendert werden soll. Dieses Attribut entspricht der *ID* des HTML-Elements, die entweder `DIV` oder `IFRAME` lauten kann. Für die Modellverwaltung verweist der Berichtspeicherort immer auf `DIV`.
- Das Attribut *outputType*, das das Format angibt, in dem der Bericht mithilfe von IBM SPSS Collaboration and Deployment Services Tag Library gerendert werden soll. Es muss ein Ausgabeformat verwendet werden, das von der BIRT Report Designer for IBM SPSS Report Engine unterstützt wird. Für die Berichte, die in der Modellverwaltung verwendet werden, ist der Ausgabetyt immer *HTML*.

Tag "sourceLinkPrompt"

Der Tag `sourceLinkPrompt` legt die Verknüpfung von Eingabeaufforderungen mit dem Bericht fest. Anders ausgedrückt legt dieser Tag fest, wie der Bericht die Eingabeaufforderungswerte bei der Ausführung abrufen.

Die Eingabeaufforderungswerte werden für die Seite "Modellverwaltung" auf zwei unterschiedliche Arten angegeben. Zum einen können sie über das Attribut *parameterValue* angegeben werden:

```
<!-- Tag, der den Bericht darstellt -->
<ibmspss-taglib:repositoryItem
  name="Performance_Tab_Report_Tag"
  inputURI="<<Pfad des Leistungsberichts>>"
  repositoryCredentialName="localhost"
  activate="ONLOAD"
  location="Performance_Tab_Report_Output"
  outputType="HTML">
  <ibmspss-taglib:sourceLinkPrompt
    targetNameParameter="LeastPerformingScenarios"
    parameterValue="<<localized text>>" />

<!--
  <<<< Sonstige Parameter, die lokalisierten Text darstellen>>>>
-->

<!-- Dieser Wert kommt aus benutzerdefinierten Eingabeaufforderungen -->
<ibmspss-taglib:sourceLinkPrompt
  promptId="Performance_Tab_NumberOfPerformers_Prompt"
  targetNameParameter="NumberOfPerformers"/>
<!-- Der Wert dieses Parameters wird direkt im Tag angegeben -->
<ibmspss-taglib:sourceLinkPrompt
  targetNameParameter="RunsFromDate"
  parameterValue="<%=scenariosFrom%>" />
<ibmspss-taglib:sourceLinkPrompt
  targetNameParameter="RunsToDate"
  parameterValue="<%=scenariosTo%>" />
</ibmspss-taglib:repositoryItem>
```

Hier lautet der Name der Eingabeaufforderung *RunsFromDate* (im Bericht definiert). Der Wert für diese Eingabeaufforderung wird im Attribut *parameterValue* angegeben. Der Wert, der in diesem Attribut weitergegeben wird, wird direkt an den Bericht weitergegeben.

Des Weiteren können Eingabeaufforderungswerte angegeben werden, indem eine Benutzereingabeaufforderung mit dem Berichtsparameter verknüpft wird. Beispiel:

```

<!-- Tag, der den Bericht darstellt -->
<ibmspss-taglib:repositoryItem
  name="Performance_Tab_Report_Tag"
  inputURI="<<Pfad des Leistungsberichts>>"
  repositoryCredentialName="localhost"
  activate="ONLOAD"
  location="Performance_Tab_Report_Output"
  outputType="HTML">
  <ibmspss-taglib:sourceLinkPrompt
    targetNameParameter="LeastPerformingScenarios"
    parameterValue="<<localized text>>" />

<!--
  <<<< Sonstige Parameter, die lokalisierten Text darstellen>>>>
-->

  <!-- Dieser Wert kommt aus benutzerdefinierten Eingabeaufforderungen -->
  <ibmspss-taglib:sourceLinkPrompt
    promptId="Performance_Tab_NumberOfPerformers_Prompt"
    targetNameParameter="NumberOfPerformers"/>
  <!-- Der Wert dieses Parameters wird direkt im Tag angegeben -->
  <ibmspss-taglib:sourceLinkPrompt
    targetNameParameter="RunsFromDate"
    parameterValue="<%=scenariosFrom%>" />
  <ibmspss-taglib:sourceLinkPrompt
    targetNameParameter="RunsToDate"
    parameterValue="<%=scenariosTo%>" />
</ibmspss-taglib:repositoryItem>

<input type="hidden" id="Performance_Tab_NumberOfPerformers_Prompt"
  name="Performance_Tab_NumberOfPerformers_Prompt"
  value="<%=userProfile.getPerformanceSize()%>" />

```

Hier verweist das Attribut *promptId* auf die ID, die vom verdeckten HTML-Eingabetag definiert wird. In diesem Fall würde der Wert, der im verdeckten Feld *Performance_Tab_NumberOfPerformers_Prompt* angegeben ist, bei der Ausführung des Berichts als Eingabeaufforderungswert für den Berichtsparameter *NumberOfPerformers* weitergegeben.

Berechtigungsachweise

Die Berichte, aus denen die Seite "Modellverwaltung" besteht, fragen ihre Inhalte in der Datenbank ab, auf der das Repository beruht. Daher benötigen die Berichte eine Datenquelle, die dieser Datenbank entspricht. Diese Datenbank, *MMDDataSource*, wird im Repository erstellt, wenn der Benutzer die Seite "Modellverwaltung" erstmals lädt, und immer verwendet, wenn für einen der Tags Zugriff auf die Repository-Datenbank erforderlich ist.

Um auf die Datenquelle *MMDDataSource* zuzugreifen, muss der Bericht gültige Berechtigungsachweise angeben. Der Tag *credential* auf den JavaServer Pages ermöglicht die Angabe dieser Berechtigungsachweise.

```

<ibmspss-taglib:credential
  name="MMDDataSource"
  username="<<Name eines Datenbankbenutzers>>"
  password="<<Kennwort des Benutzers>>" />

```

Die Berechtigungsachweise für diese Datenquelle werden über *Login.jsp* abgerufen, bevor die Seite angezeigt wird, und entsprechen dem Benutzernamen und dem Kennwort für die Datenbank, auf der das Repository basiert. Wenn gültige Berechtigungsachweise angegeben werden, werden sie für die Dauer der Sitzung im Cache gespeichert und für die Ausführung der Berichte verwendet. Die Modellverwaltungsberichte sind so definiert, dass der Name der Datenquelle *MMDDataSource* lautet.

Neben den Berechtigungsachweisen für die Datenquelle sind für Modellverwaltungsberichte auch Berechtigungsachweise für den Benutzer erforderlich, der den Bericht ausführt.

```

<ibmspss-taglib:credential
  name="localhost"
  provider="<<eine Provider-ID>>"
  username="<<Name eines CR-Benutzers>>"
  password="<<Kennwort des Benutzers>>" />

```

Diese Berechtigungsachweise haben den Namen *localhost*. Wenn das Repository mehrere Sicherheitsprovider zulässt, muss das Attribut *provider* angegeben werden.

Der Tag `repositoryItem` erfordert gültige Benutzerberechtigungsnaehweise für das Repository, die im Attribut `repositoryCredentialName` des Tags angegeben werden. Bei der Modellverwaltung ist der Wert für dieses Attribut `localhost`. Dies entspricht dem Benutzernamen, dem Provider und dem Kennwort des Benutzers, der sich bei IBM SPSS Collaboration and Deployment Services Deployment Portal angemeldet hat.

Ausführen von Visualisierungsberichten

Die Methodologie der Ausführung von Visualisierungsberichten ist mit der für Berichte von BIRT Report Designer for IBM SPSS identisch.

Es sind allerdings folgende Unterschiede bei der Verwendung zu berücksichtigen:

- Visualisierungsberichte verwenden den Wert `ONDEMAND` für das Attribut `activate` des Tags `repositoryItem` und nicht den Wert `ONLOAD`, der in Berichten von BIRT Report Designer for IBM SPSS verwendet wird.
- Parameter, die für Visualisierungsberichte erforderlich sind, werden von den Masterberichten von BIRT Report Designer for IBM SPSS weitergegeben. Weitere Informationen finden Sie im Thema „Interaktivität von Visualisierungsberichten“.

JavaScript-API

Die Tagbibliothek verfügt über ein Framework, das mit JavaScript-Methoden erstellt wurde. Diese JavaScript-Methoden bieten sowohl ein solides Validierungsframework als auch ein Handle für den Benutzer zur Ausführung von Berichten bei Bedarf.

Um die Berichte bei Bedarf auszuführen, stellt die Tagbibliothek eine öffentliche API bereit. Diese öffentliche API ist in der Datei `reportTagLibPublicAPI.js` unter `IBMSPSSTagLib.war` verfügbar. Die JavaScript-Datei enthält folgende API:

```
function runRepositoryItem( reportName, linkData, targetId )
```

Für die Modellverwaltung wird diese Funktion verwendet, um untergeordnete Berichte für den Masterbericht aufzurufen.

Wenn die Registerkarte "Szenario" angezeigt wird, werden beispielsweise die Szenarioberichtsdaten angezeigt. Wenn der Benutzer auf den Link für ein Szenario im Masterbericht klickt, wird die JavaScript-Methode `showDetails` aufgerufen. Diese JavaScript-Methode ist in den Bericht von BIRT Report Designer for IBM SPSS eingebettet und ruft indirekt die Methode `runRepositoryItem` für die Ausführung von zwei Berichten auf. Einer der Berichte ist der Bericht für die Szenariodetails von BIRT Report Designer for IBM SPSS, der andere ist der Visualisierungsbericht für den Szenariovergleich.

Wenn `linkData` im API-Aufruf null ist, wird der Bericht mit den Daten ausgeführt, die auf der JavaServer Page verfügbar sind, die über die unterschiedlichen Tags von IBM SPSS Collaboration and Deployment Services Tag Library bereitgestellt werden. Kurz vor dem Aufruf von `runRepositoryItem` speichert der JavaScript-Code die Parameterwerte im verdeckten HTML-Steuerelement. Das Tagbibliothekframework ruft diese Werte ab und gibt sie als Parameter an den ausgeführten Bericht weiter.

Die `targetID`-Felder entsprechen den einzelnen DIV-IDs, in denen der Bericht gespeichert werden soll.

Interaktivität von Visualisierungsberichten

Das Diagramm zur Leistung im Vergleich zum Szenario, das durch den Visualisierungsbericht für die Registerkarte "Champions" erzeugt wurde, unterstützt Interaktivität. Wenn der Benutzer auf einen Balken im Diagramm klickt, werden die Details des entsprechenden Szenarios in einem angrenzenden Bereich angezeigt. Die Berichte verwenden für diese Funktion die Tags `actionHandler` und `actionParameter`.

Die Verwendung des Tags `actionHandler` ist für Visualisierungsberichte nicht unbedingt erforderlich. Normalerweise würde der Tag `sourceLinkRepositoryItem` einen ebenso guten Service für Visualisierungs-

berichte erweisen. Auf der Seite "Modellverwaltung" kann das Visualisierungsdiagramm jedoch mehrmals angezeigt werden. Die Anwendung benötigt eine spezielle Logik, um Detailzeilen zu erweitern und die Zielberichte mit den jeweiligen Ausgabepositionen auszuführen. Die `actionHandler`-Tags bieten diese zusätzliche Steuerebene.

Im Folgenden finden Sie den Abschnitt der Seite, die den Visualisierungsbericht für die Leistung im Vergleich zum Szenario rendert:

```
<ibmspss-taglib:repositoryItem
name="Champions_Scenario_Index_Report"
  inputURI="ChampionsScenarioIndex.viz"
  repositoryCredentialName="localhost"
  activate="ONDEMAND"
  outputType="HTML"
  location="championsTabVisReport">
  <ibmspss-taglib:actionHandler event="onclick" function="selectCCScenario">
    <ibmspss-taglib:actionParameter name="filename" />
    <ibmspss-taglib:actionParameter name="filepath" />
    <ibmspss-taglib:actionParameter name="ccid" />
    <ibmspss-taglib:actionParameter name="equivalencekey" />
  </ibmspss-taglib:actionHandler>
</ibmspss-taglib:repositoryItem>
```

Der Tag `repositoryItem` liefert Details zu dem zu erstellenden Balkendiagramm. Der verschachtelte Tag `actionHandler` gibt an, dass die JavaScript-Funktion `selectCCScenario` aufgerufen werden soll, wenn das Ereignis `onClick` für die Balken eintritt. Die `actionParameter`-Tags, die im Tag `actionHandler` verschachtelt sind, geben an, dass *filename*, *filepath*, *ccid* und *equivalencekey* an die Funktion `selectCCScenario` weitergegeben werden.

Jedes dieser Felder wird im XML des Visualisierungsberichts definiert. Im Folgenden finden Sie die Definition für die Variable *filename*:

```
<sourceVariable
categorical="true"
id="filename"
source="delimitedFileSource_430"
sourceName="ct_filename">
```

Dieser Tag gibt an, dass die im Dataset als *ct_filename* definierte Spalte in diesem Bericht als *filename* verwendet wird.

Die JavaScript-Funktion `selectCCScenario` erhält die ID des Berichts, in dem das Ereignis eingetreten ist, und ein Array der Parameterwerte. Intern ruft es `runReport` nach abhängigen untergeordneten Berichten auf und gibt das Werte-Array an sie weiter. Weitere Informationen finden Sie im Thema „JavaScript-API“ auf Seite 61.

URL-Fragmente

Die Seite "Modellverwaltung" zeigt einige Repository-Artefakte in einem I-FRAME an. Diese Artefakte sind die Ausgaben, die von bestimmten Jobausführungen erzeugt werden.

Ein Artefakt wird geladen, indem die Quelle des I-FRAME auf die URL mit folgendem Format gesetzt wird:

```
http://<Servername>:<Port>/peb/view?id=<artifact resource id>
```

Framework "Tab Extension"

Die Navigationsregisterkarten von IBM SPSS Collaboration and Deployment Services Deployment Portal können über das Framework "Tab Extension" mit benutzerdefinierten Einträgen erweitert werden. Die Funktion "Modellverwaltung" verwendet dieses Framework, um einen Einstiegspunkt in die Seite "Modellverwaltung" zu integrieren.

IBM SPSS Collaboration and Deployment Services Deployment Portal liest Erweiterungsdateien, die sich in folgendem Verzeichnis befinden:

```
<Installationsverzeichnis>/components/peb/extensions
```

Diese Dateien werden nach allen Instanzen der `peb-extension`-Elemente durchsucht. Diese Elemente werden separat an der Schnittstelle angezeigt, wenn die Benutzerberechtigungsnaehweise erforderliche Aktionen umfassen. Benutzerdefinierte Anwendungen müssen folgende Elemente bereitstellen:

- Erweiterungs-XML-Datei oder einen Eintrag in einem vorhandenen Erweiterungs-XML für die Anwendung
- Geeignete Einträge in der lokalisierten Textdatei (.tx)

Die Funktion "Modellverwaltung" ist in der Datei `peb-mmd.package` im Verzeichnis `staging` der Repository-Installation enthalten. Das Paket enthält die Datei `mmd_extension.xml` im Verzeichnis `peb/extensions`. Diese XML-Datei steuert das Erscheinungsbild und die Funktionen der Registerkarte "Modellverwaltung".

```
<file-viewer>
  <peb-extension>
    <tab-id>pebMmdTab</tab-id>
    <tab-key>mmd/pebMmdTabTitle</tab-key>
    <tab-url>
      /peb-mmd/controller?actionName=LoginToMMDAction
    </tab-url>
    <tab-icon>/image?file=someIcon.gif</tab-icon>
    <tab-position>2</tab-position>
    <tab-security>
      <capability>RunReport</capability>
      <capability>ViewModelManagementDashboard</capability>
    </tab-security>
  </peb-extension>
</file-viewer>
```

Folgende Elemente werden in dieser Datei definiert:

- Das Element `tab-id`, welches die eindeutige ID für die Registerkarte ist. In diesem Fall lautet die ID `pebMmdTab`.
- Das Element `tab-key`, das auf den Text verweist, der auf der neuen Registerkarte angezeigt wird. Die Modellverwaltung isoliert lokalisierten Text in XML-Dateien mit der Erweiterung `.tx`. Der Schlüssel kennzeichnet das Element in der Datei mit sprachabhängigen Anweisungen, das den anzuzeigenden Text enthält. In diesem Fall entspricht der Schlüssel `mmd/pebMmdTabTitle` dem Text *Modellverwaltung*.
- Das Element `tab-url`, das die URL angibt, die aufgerufen wird, wenn der Benutzer auf die Registerkarte klickt. Die URL kann entweder vollständig qualifiziert (mit einem Schrägstrich '/' beginnend) oder relativ zur Anwendung IBM SPSS Collaboration and Deployment Services Deployment Portal angegeben werden. Im letzteren Fall wird `peb` als Kontext angenommen. Der Link muss auf einen gültigen URI verweisen, wobei der angegebene URI-Speicherort in der Verantwortung der benutzerdefinierten Anwendung liegt. Für die Modellverwaltung enthält der Link einen Verweis auf die WAR-Datei `peb-mmd.war`.
- Das Element `tab-security`, das die Aktionen festlegt, die für den Zugriff auf die Registerkarte erforderlich sind. Wenn der aktuelle Benutzer diese Aktionen nicht ausführt, wird die Registerkarte nicht auf der Header-JSP angezeigt. Für die Modellverwaltung sind die Aktionen `RunReport` und `ViewModelManagementDashboard` erforderlich.

Kapitel 7. Erstellen von benutzerdefinierten Datenservicetreibern

Die DataService-API von IBM SPSS Collaboration and Deployment Services umfasst öffentliche Java-Schnittstellen zum Implementieren benutzerdefinierter Treiber, um auf vom Standard abweichende Datenquellen zugreifen zu können.

Ein benutzerdefinierter Treiber kann z. B. erforderlich sein, wenn aus Sicherheitsgründen kein JDBC- oder ODBC-Zugriff auf Datenbankumgebungen für die Produktion zulässig ist. Ein solcher Treiber kann auch für dateibasierte Datenquellen, Daten im Cache oder ältere Datenquellen erforderlich sein, die von JDBC bzw. ODBC nicht unterstützt werden. Diese Funktion wird überwiegend vom Scoring-Service verwendet.

In diesem Anhang finden Sie einen Überblick über die DataService-API und Anweisungen zum Erstellen eines benutzerdefinierten Treibers sowie ein Beispiel für einen benutzerdefinierten Treiber, der im Umfang von IBM SPSS Collaboration and Deployment Services enthalten ist.

Datenservice-API

Die API zum Implementieren von benutzerdefinierten Treibern ist im Paket *com.spss.data.service.datasource* als eine Gruppe von Schnittstellen enthalten.

- *com.spss.data.service.datasource.DataService*
- *com.spss.data.service.datasource.DataServiceSession*
- *com.spss.data.service.datasource.DataServiceRequest*

Die Paketklassen befinden sich in der Datei *data-service.jar* des Verzeichnisses *<Repository-Installationsverzeichnis>/staging/scoring.package*.

Initialisierung

DataService-Objekte werden zum Initialisieren von mindestens einer Verbindungsinstanz mit den Parametern verwendet, die für die Datenquelle definiert wurden, z. B. Name und Eigenschaften der Datenquelle. Bei der Klasse, die zum Implementieren der *DataService*-Schnittstelle verwendet wird, muss der Standardkonstruktor verwendet werden.

Verbindungsinstanzen werden über mehrere Ereignisse im Cache abgelegt. Ressourcenintensive Initialisierungen und das Ablegen von Ressourcen im Cache können durchgeführt werden, da die Anzahl instanzierter Verbindungen in der Regel klein ist. Für dieselbe Datenquelle können mehrere Instanzen vorliegen. Es empfiehlt sich, Systemressourcen mit der Methode *terminate* zu veröffentlichen. Eine einzelne Verbindungsinstanz muss in der Lage sein, gleichzeitig mehrere Aufrufe verarbeiten zu können. Bei jedem Aufruf wird ein eigenes *DataServiceRequest*-Objekt erstellt, das für ein einzelnes Abrufen von Daten verwendet werden kann. Dieses Objekt kann nur einmal abgefragt werden. Die Datenserviceverbindung muss die Abfrageergebnisse nicht im Cache ablegen. Bei einzelnen Ereignissen wird der DataService mit einer bestimmten Mindesthäufigkeit aufgerufen. Beachten Sie, dass zirkuläre Links dazu führen können, dass Tabellen mehrmals abgefragt werden. Die Datenserviceverbindung sollte Fehler mittels *DataServiceException* melden. Andere Ausnahmen sind nicht zulässig.

Datenverarbeitung

Das *DataServiceSession*-Objekt wird während der Verarbeitung eines Ereignisses von *DataService* instanziiert und das Objekt kann dazu verwendet werden, um einen ereignisbasierten Status der Sitzung beizubehalten. Das Sitzungsobjekt instanziiert dann mit der Methode *createRequest* ein *DataServiceRequest*-Objekt für jede Abfrage mit den folgenden Parametern:

- dem Namen der Tabelle.
- den Namen der Attribute (Spalten), für die die Werte zurückgegeben werden sollen.
- der Liste der Schlüsselattributnamen.
- Schlüsseltupeln, die als eine Reihe von Objektlisten angegeben werden. Die Reihenfolge der Objekte in den Listen stimmt mit der Reihenfolge der Attribute überein, die in den Schlüsselattributnamen angegeben sind.

Mit *DataServiceRequest* wird eine Liste der Datensätze zurückgegeben. Jeder Datensatz umfasst eine Liste mit Attributwerten. Die Liste der Attributwerte muss mit der Liste der Attributnamen übereinstimmen, die an die Methode *createRequest* übergeben wurde. Attributwerte werden als Java-Objektinstanzen zurückgegeben. Für jeden zurückgegebenen Wert müssen die mit Attributen übereinstimmenden Wertesätze, die Bestandteil des Schlüssels sind, mit einem der Schlüsseltupel übereinstimmen. Im Folgenden werden die Zuordnungen zwischen Datentypen von IBM SPSS Collaboration and Deployment Services Enterprise View und Java-Klassen aufgeführt:

Datentyp von IBM SPSS Collaboration and Deployment Services Enterprise View	Erforderliche Java-Klasse
Ganzzahl	java.lang.Integer
Lang	java.lang.Long
Double	java.lang.Double
Gleitkomma	java.lang.Float
Zeichenfolge	java.lang.String
Datum	java.sql.Date
Tageszeit	java.sql.Time
Zeitmarke	java.sql.Timestamp
Dezimal	java.math.BigDecimal
Boolesch	java.lang.Boolean

Zur Rückgabe eines *Nil*-Objekts mit `value.isNil() = true` muss das Nullobjekt zurückgegeben werden.

Erstellen eines benutzerdefinierten Treibers

So erstellen Sie einen benutzerdefinierten Treiber mit der *Datenservice*-API:

1. Extrahieren Sie *data-server.jar* aus *scoring.package* und stellen Sie sicher, dass sich das Objekt im Klassenpfad befindet.
2. Schreiben Sie den Java-Quellcode, mit dem die *Datenserviceschnittstellen* implementiert werden, und kompilieren Sie die Treiberklassen.
3. Erstellen Sie das Treiberpaket (als JAR-Datei).
4. Stellen Sie das Paket im Verzeichnis `<Repository-Installationsverzeichnis>/components/data-provider` der Installation von IBM SPSS Collaboration and Deployment Services bereit. Informationen zu geclusterten Installationen finden Sie im Abschnitt zur Clusterbildung in der Installations- und Konfigurationsdokumentation.
5. Starten Sie je nach Anwendungsserver IBM SPSS Collaboration and Deployment Services neu.

Erstellen Sie zum Testen des Treibers eine entsprechende Datenquelle in Deployment Manager, fügen Sie sie zur Enterprise View hinzu und erstellen Sie eine Vorschau der Daten.

Beispiel eines benutzerdefinierten Treibers

Die Installation von IBM SPSS Collaboration and Deployment Services beinhaltet ein Beispiel eines benutzerdefinierten Treibers. Die Java-Quellendateien für das Beispiel finden Sie unter <Repository-Installationsverzeichnis>/help/custom_driver.

Dazu gehören:

- *WSDataServiceConnector.java* (implementiert *com.spss.data.service.dataservice.DataService*)
- *WSDataServiceRequest.java* (implementiert *com.spss.data.service.dataservice.DataServiceRequest*)
- *WSDataServiceSession.java* (implementiert *com.spss.data.service.dataservice.DataServiceSession*)

Die Beispieldateien enthalten zudem die ausführbare Datei *sample.jar*.

Der Treiber ruft die Daten aus der Textdatei *DATA_1* ab. Dabei wird der Wert des in der Datenservice-Datenquelle angegebenen Schlüssels zugrunde gelegt. Das Dateiformat lautet wie folgt:

```
data_key=1  
data_element_0=Dies sind die Daten des ersten Elements  
data_element_1=Dies sind die Daten des zweiten Elements
```

So können Sie die Funktionsfähigkeit des Treibers demonstrieren:

1. Stellen Sie *sample.jar* bereit.
2. Erstellen Sie eine Datenservice-Datenquelle mit folgenden Parametern:
 - **com.spss.rts.beil.ds.connector.WSDataServiceConnector** als Name der Treiberklasse.
 - Treibereigenschaft **DataLocation**, die auf das Verzeichnis verweist, in dem sich die Datendatei *DATA_1* befindet.
3. Definieren Sie eine Tabelle für die Datenquelle mit folgenden Spalten:
 - **data_element_0** (Zeichenfolge)
 - **data_element_1** (Zeichenfolge)
 - **data_key** (Zeichenfolge, muss als primärer Schlüssel gekennzeichnet sein)
4. Erstellen Sie die Datendatei *DATA_1* wie angegeben und stellen Sie sicher, dass sie sich an dem durch die Eigenschaft *DataLocation* des Treibers angegebenen Speicherort befindet.
5. Erstellen Sie eine Echtzeit-Datenproviderdefinitionen.
6. Fügen Sie mithilfe der Registerkarte **Datasets** des Echtzeiteditors für die Datenproviderdefinition die in Ihrer Datenservice-Datenquelle definierte Tabelle zur Datenproviderdefinition hinzu.
7. Zeigen Sie eine Vorschau der Daten in der Tabelle an. Geben Sie dabei 1 als Schlüsselwert an. Auf dem Ergebnisbildschirm werden die Inhalte der dem Schlüsselwert entsprechenden Datendatei angezeigt.

Bemerkungen

Die vorliegenden Informationen wurden für Produkte und Services entwickelt, die auf dem deutschen Markt angeboten werden.

Möglicherweise bietet IBM die in dieser Dokumentation beschriebenen Produkte, Services oder Funktionen in anderen Ländern nicht an. Informationen über die gegenwärtig im jeweiligen Land verfügbaren Produkte und Services sind beim zuständigen IBM Ansprechpartner erhältlich. Hinweise auf IBM Lizenzprogramme oder andere IBM Produkte bedeuten nicht, dass nur Programme, Produkte oder Services von IBM verwendet werden können. Anstelle der IBM Produkte, Programme oder Services können auch andere, ihnen äquivalente Produkte, Programme oder Services verwendet werden, solange diese keine gewerblichen oder anderen Schutzrechte von IBM verletzen. Die Verantwortung für den Betrieb von Produkten, Programmen und Services anderer Anbieter liegt beim Kunden.

Für in diesem Handbuch beschriebene Erzeugnisse und Verfahren kann es IBM Patente oder Patentanmeldungen geben. Mit der Auslieferung dieses Handbuchs ist keine Lizenzierung dieser Patente verbunden. Lizenzanforderungen sind schriftlich an folgende Adresse zu richten (Anfragen an diese Adresse müssen auf Englisch formuliert werden):

IBM Director of Licensing
IBM Europe, Middle East & Africa
Tour Descartes
2, avenue Gambetta
92066 Paris La Defense
France

Diese Informationen können technische Ungenauigkeiten oder typografische Fehler enthalten. Die hier enthaltenen Informationen werden in regelmäßigen Zeitabständen aktualisiert und als Neuausgabe veröffentlicht. IBM kann ohne weitere Mitteilung jederzeit Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen.

Verweise in diesen Informationen auf Websites anderer Anbieter werden lediglich als Service für den Kunden bereitgestellt und stellen keinerlei Billigung des Inhalts dieser Websites dar. Das über diese Websites verfügbare Material ist nicht Bestandteil des Materials für dieses IBM Produkt. Die Verwendung dieser Websites geschieht auf eigene Verantwortung.

Werden an IBM Informationen eingesandt, können diese beliebig verwendet werden, ohne dass eine Verpflichtung gegenüber dem Einsender entsteht.

Lizenznehmer des Programms, die Informationen zu diesem Produkt wünschen mit der Zielsetzung: (i) den Austausch von Informationen zwischen unabhängig voneinander erstellten Programmen und anderen Programmen (einschließlich des vorliegenden Programms) sowie (ii) die gemeinsame Nutzung der ausgetauschten Informationen zu ermöglichen, wenden sich an folgende Adresse:

IBM Software Group
ATTN: Licensing
200 W. Madison St.
Chicago, IL; 60606
USA

Die Bereitstellung dieser Informationen kann unter Umständen von bestimmten Bedingungen - in einigen Fällen auch von der Zahlung einer Gebühr - abhängig sein.

Die Lieferung des in diesem Dokument beschriebenen Lizenzprogramms sowie des zugehörigen Lizenzmaterials erfolgt auf der Basis der IBM Rahmenvereinbarung bzw. der Allgemeinen Geschäftsbedingungen von IBM, der IBM Internationalen Nutzungsbedingungen für Programmpakete oder einer äquivalenten Vereinbarung.

Alle in diesem Dokument enthaltenen Leistungsdaten stammen aus einer kontrollierten Umgebung. Die Ergebnisse, die in anderen Betriebsumgebungen erzielt werden, können daher erheblich von den hier erzielten Ergebnissen abweichen. Einige Daten stammen möglicherweise von Systemen, deren Entwicklung noch nicht abgeschlossen ist. Eine Gewährleistung, dass diese Daten auch in allgemein verfügbaren Systemen erzielt werden, kann nicht gegeben werden. Darüber hinaus wurden einige Daten unter Umständen durch Extrapolation berechnet. Die tatsächlichen Ergebnisse können davon abweichen. Benutzer dieses Dokuments sollten die entsprechenden Daten in ihrer spezifischen Umgebung prüfen.

Alle Informationen zu Produkten anderer Anbieter stammen von den Anbietern der aufgeführten Produkte, deren veröffentlichten Ankündigungen oder anderen allgemein verfügbaren Quellen. IBM hat diese Produkte nicht getestet und kann daher keine Aussagen zu Leistung, Kompatibilität oder anderen Merkmalen machen. Fragen zu den Leistungsmerkmalen von Produkten anderer Anbieter sind an den jeweiligen Anbieter zu richten.

Aussagen über Pläne und Absichten von IBM unterliegen Änderungen oder können zurückgenommen werden und repräsentieren nur die Ziele von IBM.

Diese Veröffentlichung enthält Beispiele für Daten und Berichte des alltäglichen Geschäftsablaufs. Sie sollen nur die Funktionen des Lizenzprogramms illustrieren und können Namen von Personen, Firmen, Marken oder Produkten enthalten. Alle diese Namen sind frei erfunden; Ähnlichkeiten mit tatsächlichen Namen und Adressen sind rein zufällig.

COPYRIGHTLIZENZ:

Diese Veröffentlichung enthält Beispielanwendungsprogramme, die in Quellsprache geschrieben sind und Programmier Techniken in verschiedenen Betriebsumgebungen veranschaulichen. Sie dürfen diese Beispielprogramme kostenlos kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, zu verwenden, zu vermarkten oder zu verteilen, die mit der Anwendungsprogrammierschnittstelle für die Betriebsumgebung konform sind, für die diese Beispielprogramme geschrieben werden. Diese Beispiele wurden nicht unter allen denkbaren Bedingungen getestet. Daher kann IBM die Zuverlässigkeit, Wartungsfreundlichkeit oder Funktion dieser Programme weder zusagen noch gewährleisten. Die Beispielprogramme werden ohne Wartung (auf "as-is"-Basis) und ohne jegliche Gewährleistung zur Verfügung gestellt. IBM übernimmt keine Haftung für Schäden, die durch die Verwendung der Beispielprogramme entstehen.

Hinweise zur Datenschutzrichtlinie

IBM Softwareprodukte, einschließlich Software as a Service-Lösungen ("Softwareangebote"), können Cookies oder andere Technologien verwenden, um Informationen zur Produktnutzung zu erfassen, die Endbenutzererfahrung zu verbessern und Interaktionen mit dem Endbenutzer anzupassen oder zu anderen Zwecken. In vielen Fällen werden von den Softwareangeboten keine personenbezogenen Daten erfasst. Einige der IBM Softwareangebote können Sie jedoch bei der Erfassung personenbezogener Daten unterstützen. Wenn dieses Softwareangebot Cookies zur Erfassung personenbezogener Daten verwendet, sind nachfolgend nähere Informationen über die Verwendung von Cookies durch dieses Angebot zu finden.

Dieses Softwareangebot verwendet keine Cookies oder andere Technologien zur Erfassung personenbezogener Daten.

Wenn die für dieses Softwareangebot genutzten Konfigurationen Sie als Kunde in die Lage versetzen, personenbezogene Daten von Endbenutzern über Cookies und andere Technologien zu erfassen, müssen

Sie sich zu allen gesetzlichen Bestimmungen in Bezug auf eine solche Datenerfassung, einschließlich aller Mitteilungspflichten und Zustimmungsanforderungen, rechtlich beraten lassen.

Weitere Informationen zur Nutzung verschiedener Technologien, einschließlich Cookies, für diese Zwecke finden Sie in der "IBM Online-Datenschutzerklärung, Schwerpunkte" unter <http://www.ibm.com/privacy>, in der "IBM Online-Datenschutzerklärung" unter <http://www.ibm.com/privacy/details> im Abschnitt "Cookies, Web-Beacons und sonstige Technologien" und in "IBM Software Products and Software-as-a-Service Privacy Privacy Statement" unter <http://www.ibm.com/software/info/product-privacy>.

Marken

IBM, das IBM Logo und [ibm.com](http://www.ibm.com) sind Marken oder eingetragene Marken der IBM Corporation in den USA und/oder anderen Ländern. Weitere Produkt- und Servicenamen können Marken von IBM oder anderen Unternehmen sein. Eine aktuelle Liste der IBM Marken finden Sie auf der Webseite "Copyright and trademark information" unter www.ibm.com/legal/copytrade.shtml.

Adobe, das Adobe-Logo, PostScript und das PostScript-Logo sind Marken oder eingetragene Marken der Adobe Systems Incorporated in den USA und/oder anderen Ländern.

Intel, das Intel-Logo, Intel Inside, das Intel Inside-Logo, Intel Centrino, das Intel Centrino-Logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium und Pentium sind Marken oder eingetragene Marken der Intel Corporation oder ihrer Tochtergesellschaften in den USA oder anderen Ländern.

Linux ist eine eingetragene Marke von Linus Torvalds in den USA und/oder anderen Ländern.

Microsoft, Windows, Windows NT und das Windows-Logo sind Marken von Microsoft Corporation in den USA und/oder anderen Ländern.

UNIX ist eine eingetragene Marke von The Open Group in den USA und anderen Ländern.

Java und alle auf Java basierenden Marken und Logos sind Marken oder eingetragene Marken der Oracle Corporation und/oder ihrer verbundenen Unternehmen.

Weitere Produkt- und Servicenamen können Marken von IBM oder anderen Unternehmen sein.

Index

A

actionHandler (Tag) 44, 61
actionParameter (Tag) 46, 61
activate (Attribut)
 report (Tag) 58
 repositoryItem (Tag) 32
Aktionen 1
allowDownload (Parameter)
 in URL-Abfragen 17
allowPivoting (Parameter)
 in URL-Abfragen 17
allowPrinterFriendly (Parameter)
 in URL-Abfragen 17
Ausgabe
 für benutzerdefinierte Dialogfelder 24
 für Berichte 23
 für Jobs 23
 für Scoring-Modelle 24

B

batch_type (Parameter)
 in URL-Abfragen 12
Benutzerdefinierte Dialogfelder 1, 24
Benutzerdefinierter Treiber 65, 66
 Beispiel 67
Berechtigungsanzeige 1
Berichte 23
BIRT-Berichte
 JSP-Beispiele 50

C

columnName (Attribut)
 sourceLinkVariable (Tag) 44
contentType (Attribut)
 page-Aufbauregel 25
Cookies 52
credential (Tag) 29, 60
Credential-Bean 46
credentialDefinitionName (Tag)
 credential (Tag) 29

D

dataset (Parameter)
 in URL-Abfragen 11
dataset_label (Parameter)
 in URL-Abfragen 11
dataset.prompt (Parameter)
 in URL-Abfragen 13
dataset_rowlimit (Parameter)
 in URL-Abfragen 12
dataset.search.criteria (Parameter)
 in URL-Abfragen 14
dataset_table (Parameter)
 in URL-Abfragen 11

dataset.table (Parameter)
 in URL-Abfragen 13
dataset.uri (Parameter)
 in URL-Abfragen 13
Datenquelle 65, 66
Datenservice 65
Datenservice-API 65, 66
Datenservice-Datenquelle 67
dbcredential_datasourcename (Parameter)
 in URL-Abfragen 8
dbpwd_datasourcename (Parameter)
 in URL-Abfragen 9
dbuser_datasourcename (Parameter)
 in URL-Abfragen 9

E

Echtzeit-DPD 67
Eingabeaufforderungen
 für benutzerdefinierte Dialogfelder 24
 für Berichte 23
 für Jobs 23
 für Scoring-Modelle 24
emf-Dateien 6
Erstellung
 HTMLC-Dateien 55
event (Attribut)
 actionHandler (Tag) 44
Excel-Dateien 6

F

format (Parameter)
 in URL-Abfragen 8
fragment (Parameter)
 in URL-Abfragen 8
function (Attribut)
 actionHandler (Tag) 44

G

getBookmarkedValues (Funktion) 28
getValueJSFunction (Attribut)
 sourceLinkPrompt (Tag) 40

H

height (Attribut)
 repositoryItem (Tag) 32
height (Parameter)
 in URL-Abfragen 10
HTMLC-Dateien
 Erstellung 55
 Struktur 55

I

IBM SPSS Statistics, benutzerdefinierte Dialogfelder 1
IBM SPSS Statistics, Datendateitreiberservice 1
IBM SPSS Statistics-Server 1
id (Parameter)
 in URL-Abfragen 4
inputURI (Attribut)
 repositoryItem (Tag) 32

J

Java-Schnittstellen 65, 66
javascript.name (Parameter)
 in URL-Abfragen 16
javascript.url (Parameter)
 in URL-Abfragen 15
Jobs 23
jpeg-Dateien 6
JSP-Beispiele
 Zugriff 50
JSR 168 51

K

Klassenlader
 für benutzerdefinierte Anwendungen 25
 Reihenfolge 25
 Richtlinie 25

L

language (Attribut)
 page-Aufbauregel 25
linkType (Attribut)
 sourceLinkRepositoryItem (Tag) 43
localhost (Berechtigungsanzeige) 60
location (Attribut)
 outputLocation (Tag) 38
 report (Tag) 58
 repositoryItem (Tag) 32
Lotus-Dateien 6

M

MMDDDataSource 60

N

Name (Attribut)
 actionParameter (Tag) 46
 credential (Tag) 29
 repositoryItem (Tag) 32, 58

O

output.filename (Parameter)
in URL-Abfragen 16
output.format (Parameter)
in URL-Abfragen 16
outputId (Attribut)
outputLocation (Tag) 38
outputLocation (Tag) 38
outputType (Attribut)
report (Tag) 58
repositoryItem (Tag) 32
outputtype (Parameter)
in URL-Abfragen 6

P

page-Aufbauregel 25
parameterName (Attribut)
repositoryItemPrompt (Tag) 37
parameterValue (Attribut)
sourceLinkPrompt (Tag) 40, 59
partId (Attribut)
actionHandler (Tag) 44
outputLocation (Tag) 38
partId (Parameter)
in URL-Abfragen 6
password (Attribut)
credential (Tag) 29
password (Parameter)
in URL-Abfragen 5
PDF-Dateien 6
PevMetaDataBean-Bean 48
png-Dateien 6
Portal 51
Single Sign-on 52
Portlet 51
Portletpakete 51
PostScript-Dateien 6
PowerPoint-Dateien 6
prefix (Attribut)
taglib-Aufbauregel 25
promptId (Attribut)
repositoryItemPrompt (Tag) 37
sourceLinkPrompt (Tag) 40, 59
promptstate (Parameter)
in URL-Abfragen 6
provider (Attribut)
credential (Tag) 29
provider (Parameter)
in URL-Abfragen 5

R

report (Tag) 38
ReportBean-Bean 47
reportDefinitionURI (Attribut)
report (Tag) 58
reportPrompt (Tag) 38
Repository-Elemente 22
benutzerdefinierte Dialogfelder 24
Berichte 23
Jobs 23
Scoring-Modelle 24
repositoryCredentialName (Attribut)
repositoryItem (Tag) 32, 60
repositoryItem (Tag) 32, 58

repositoryItemName (Attribut)
repositoryItemPrompt (Tag) 37
repositoryItemPrompt (Tag) 37
retrievePromptValues (Funktion) 28
runRepositoryItem 61
runRepositoryItem (Funktion) 27

S

Safari-Browser 52
scoring_configuration, Parameter
in URL-Abfragen 12
Scoring-Modelle 24
ScoringBean-Bean 49
SearchBean-Bean 47
session (Attribut)
page-Aufbauregel 25
showLogs (Parameter)
in URL-Abfragen 18
showNavigationBar (Attribut)
repositoryItem (Tag) 32
showOutline (Parameter)
in URL-Abfragen 17
showTitle (Attribut)
repositoryItem (Tag) 32
showToolBar (Attribut)
repositoryItem (Tag) 32
Single Sign-on 52
sourceLinkPrompt (Tag) 40, 59
sourceLinkReport (Tag) 44
sourceLinkRepositoryItem (Tag) 43
sourceLinkVariable (Tag) 44
sourceName (Attribut)
sourceLinkRepositoryItem (Tag) 43
statistics.server (Parameter)
in URL-Abfragen 18
statistics.server.credential (Parameter)
in URL-Abfragen 18
stylesheet.name (Parameter)
in URL-Abfragen 15
stylesheet.url (Parameter)
in URL-Abfragen 15

T

taglib-Aufbauregel 25
targetNameParameter (Attribut)
sourceLinkPrompt (Tag) 40
sourceLinkVariable (Tag) 44
title (Attribut)
repositoryItem (Tag) 32

U

uri (Attribut)
taglib-Aufbauregel 25
URL-Parameter
Beispiel 62
username (Attribut)
credential (Tag) 29
username (Parameter)
in URL-Abfragen 5
useSSO (Attribut)
credential (Tag) 29

V

validate.method (Parameter)
in URL-Abfragen 16
validateJSFunction (Attribut)
sourceLinkPrompt (Tag) 40
var_variable (Parameter)
in URL-Abfragen 10
variable.display (Parameter)
in URL-Abfragen 14
variable.sort (Parameter)
in URL-Abfragen 15
Variablenparameter
in URL-Abfragen 8
version (Parameter)
in URL-Abfragen 4
Visualisierungsberichte
Interaktivität 61
JSP-Beispiele 50

W

waitstate (Parameter)
in URL-Abfragen 6
WAR-Datei 25
Web Part 51
WebSphere 25
width (Attribut)
repositoryItem (Tag) 32
width (Parameter)
in URL-Abfragen 9
Word-Dateien 6

