servBIRD 4.0

Handbuch für Anwendungsentwickler

Dokumentversion vom 20. 11. 2023

Inhalt

1 Über diese Dokumentation

1.1 Änderungshistorie

Datum	Version	Änderungsbeschreibung	Änderung von
19.08.2016	v0.1	Initialer Stand der servBIRD 3.6 Dokumentation	Team TRADUI
24.02.2017	v0.2	Beschreibung neuer Bereiche	Team TRADUI
11.12.2017	v0.3	Stand der servBIRD 3.8 Dokumentation	Team TRADUI
16.01.2019	v0.4	Stand der servBIRD 3.10 Dokumentation	Team TRADUI
12.03.2019	v0.5	• Kleinere Änderung an der Struktur • Übersetzungen nachgezogen	Busch, F. (TRADUI)
31.07.2019	v0.6	Stand der servBIRD 3.12 Dokumentation	Team TRADUI
24.06.2020	v0.7	Initialer Stand der servBIRD 3.14 Dokumentation	Team TRADUI
07.12.2020	v0.8	Initialer Stand der servBIRD 3.16 Dokumentation	Team TRADUI
10.06.2021	v0.9	Initialer Stand der servBIRD 3.18 Dokumentation	Team TRADUI
18.03.2022	v1.0	Initialer Stand der servBIRD 4.0 Dokumentation	Team TRADUI

1.2 Zielgruppe und Voraussetzungen

Die vollständige **servBIRD** Dokumentation ist in vier Handbücher aufgeteilt:

1. Das Benutzerhandbuch ...ist für den Endanwender gedacht.
2. Das Administrationshandbuch ...befasst sich mit der Installation, der Einrichtung, der Wartung und dem Betrieb des servBIRD. Dieses Handbuch ist mit seinen technischen Details für den Administrator bzw. erfahrenen Anwender gedacht.
3. Das Handbuch für Berichtsentwickler ...beleuchtet spezielle Features und Möglichkeiten von servBIRD, die im BIRT-Bericht eingestellt werden müssen.
4. Das Handbuch für Anwendungsentwickler ...ist primär für Anwendungsentwickler gedacht, die den servBIRD erweitern oder in ihre Anwendung integrieren möchten.

1.3 Hinweise und Konventionen

Tipp

Dieser Block hebt Informationen hervor, die bspw. zeit- oder ressourcenschonende Verfahren oder Best Practices erläutern.

Information

In diesem Block werden Informationen von besonderer Bedeutung oder besonderem Interesse hervorgehoben.

Hinweis

Dieser Block weist auf zu beachtende Informationen hin oder warnt vor Stolperfallen. Bitte lesen Sie diese Blöcke aufmerksam!

Achtung

Diese Hinweise deuten auf Informationen hin, die bei Missachtung oder falscher Nutzung zu Fehlfunktion der Software oder Löschung von relevanten Daten führen können.

Beispiel

In diesem Block wird ein Beispiel zum jeweiligen Kontext aufgeführt.

2 Architektur und Schnittstellen

TRADUI Technologies setzt mit **servBIRD** auf zukunftssichere offene Standards auf der Basis einer Java Enterprise Architektur.

In **servBIRD** kommen Java EE7, JSF (Java Server Faces), JPA (Java Persistence API) und OSGI-Frameworks (Open Services Gateway initiative) zum Einsatz.

servBIRD verwendet eine sogenannte Pure-Java-Structure, so **servBIRD** auf allen Betriebssystemen läuft, welche die Java EE7-Standards vollumfänglich unterstützen.

servBIRD besitzt sowohl eine SOAP- als auch eine REST-Schnittstelle.

3 servBIRD Anbindung

3.1 mooBIRD REST Webservice

servBIRD stellt einen REST Webservice für die Integration von **servBIRD** Funktionen in andere Anwendungen (APP, Portal, etc.) zur Verfügung.
Der Name der REST Schnittstelle lautet "mooBIRD".
mooBIRD wird als zusätzliches WAR File für **servBIRD** bereitgestellt.

3.1.1 Technische Dokumentation

Die technische Dokumentation der Funktionen ist in der Anwendung selbst integriert und wird unter folgender URL bereitgestellt: http://server:port/moobird

Funktionen

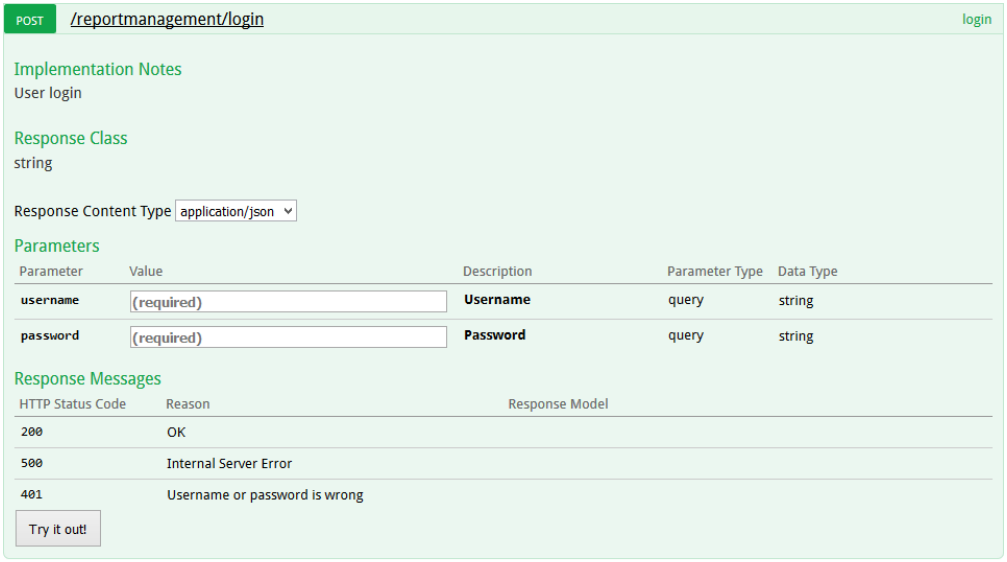
Derzeit stehen folgende Funktionen zur Verfügung:

reportmanagement : Operations about reports — Show/Hide | List Operations | Expand Operations | Raw

POST	/reportmanagement/addParameterSet/{reportfolder}/{reportdesignfile}/{parametersetname}	addParameterSet
GET	/reportmanagement/getDownloadOutputLink/{jobid}	getDownloadOutputLink
GET	/reportmanagement/getErrorOutput/{jobid}	getErrorOutput
GET	/reportmanagement/getNativeOutput/{jobid}/{native}	getNativeOutput
GET	/reportmanagement/getParameterSet/{paramsetid}	getParameterSet
GET	/reportmanagement/getParameterSet/{reportfolder}/{reportdesignfile}/{paramsetname}	getParameterSet
GET	/reportmanagement/getReportJob/{jobid}	getReportJob
GET	/reportmanagement/getReportJobStatus/{jobid}	getReportJobStatus
GET	/reportmanagement/getReportParameter	getReportParameters
GET	/reportmanagement/getReportParameterById	getReportParameters
GET	/reportmanagement/getUserReports	getUserReports
POST	/reportmanagement/login	login
POST	/reportmanagement/logout	logout
POST	/reportmanagement/startReport	startReport

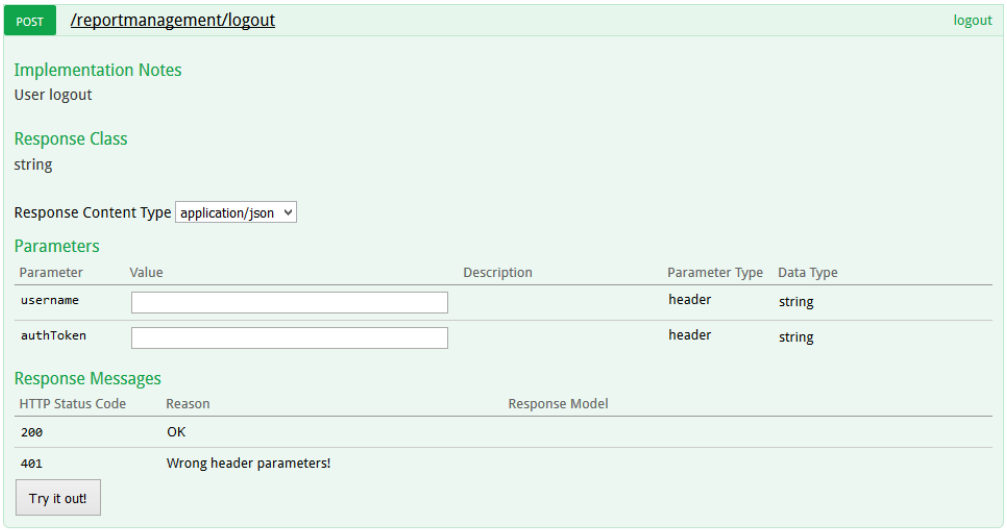
Login

Benutzer authentifizieren



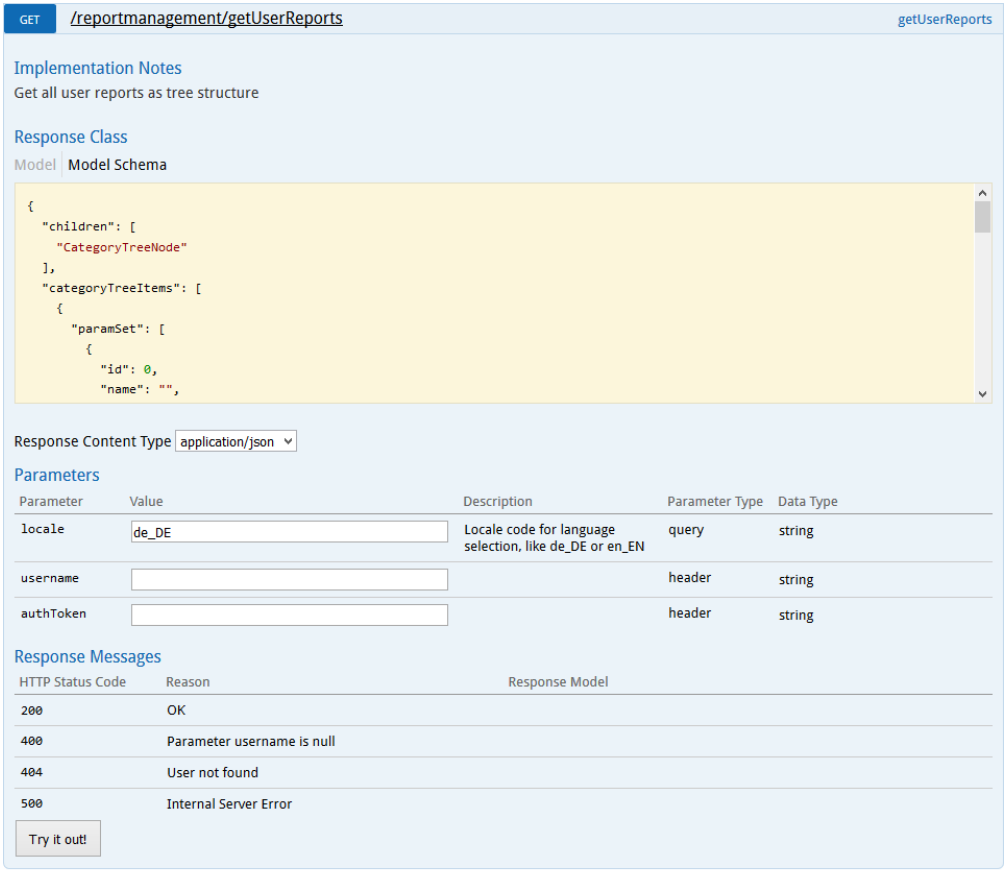
Logout

Benutzer abmelden



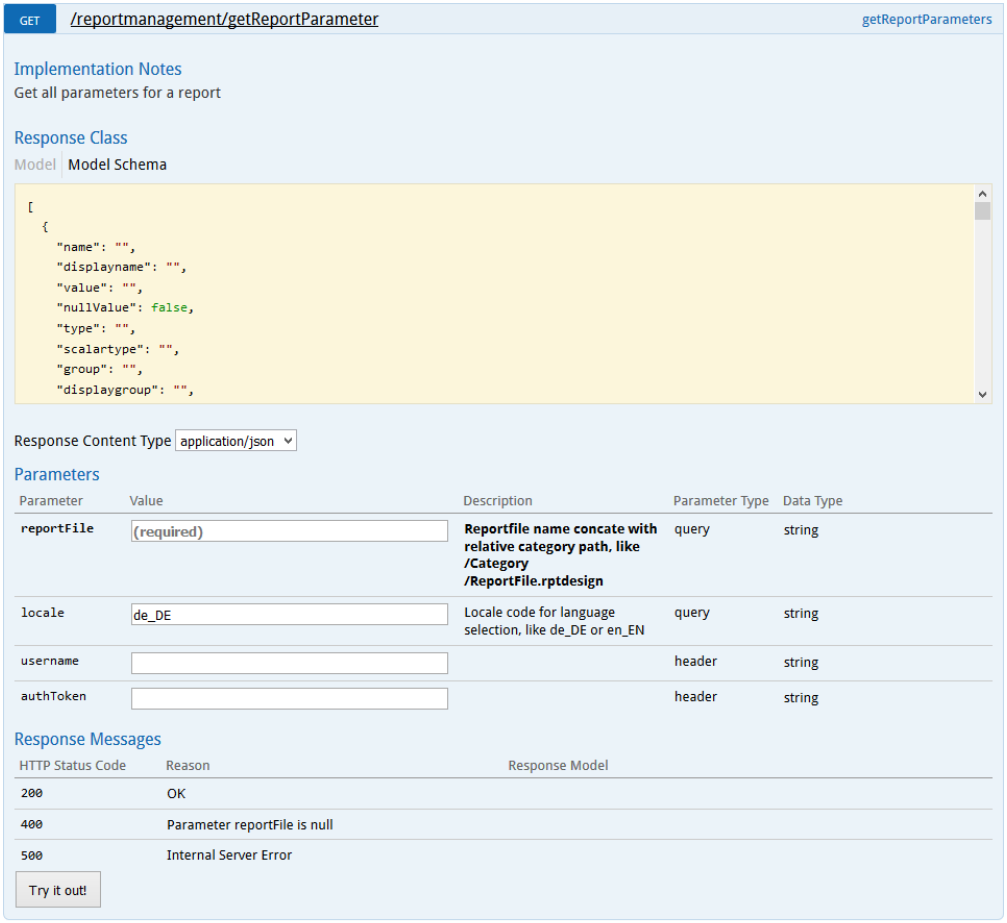
getUserReports

Kategorien, deren Berichte, Parameter und Parameter-Sets



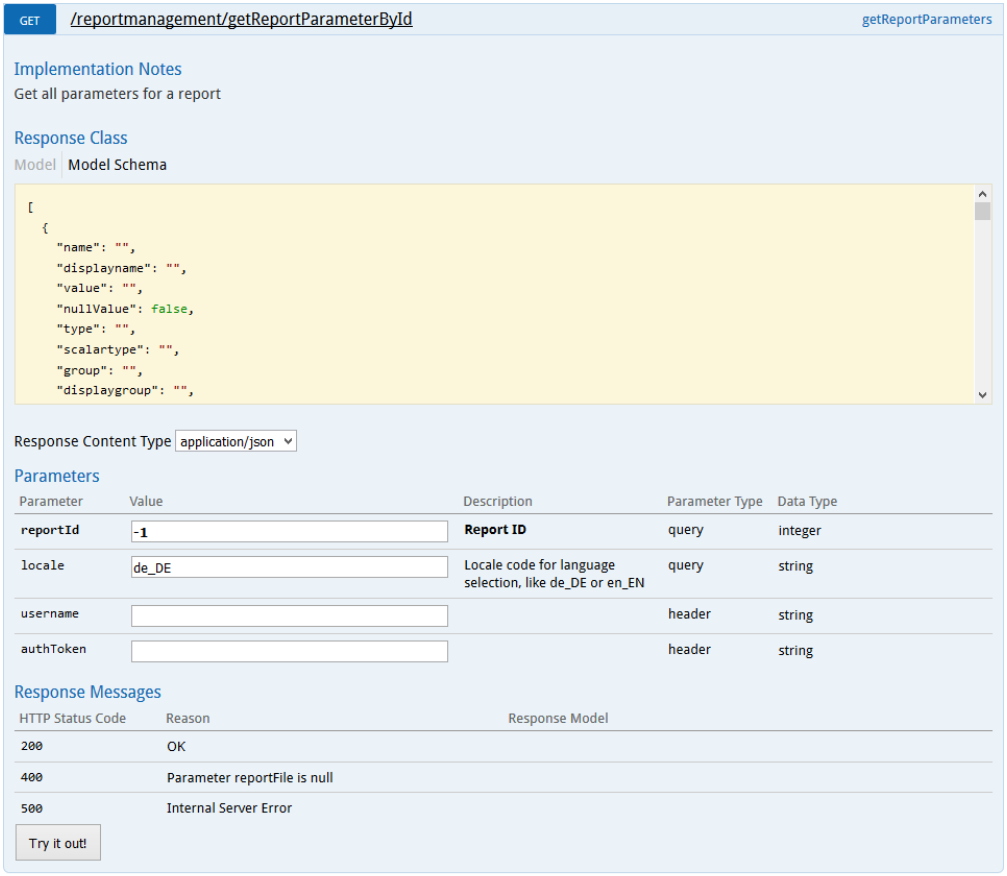
getReportParameter

Alle Parameter zu einem Bericht (Parameter Berichtsdesign)



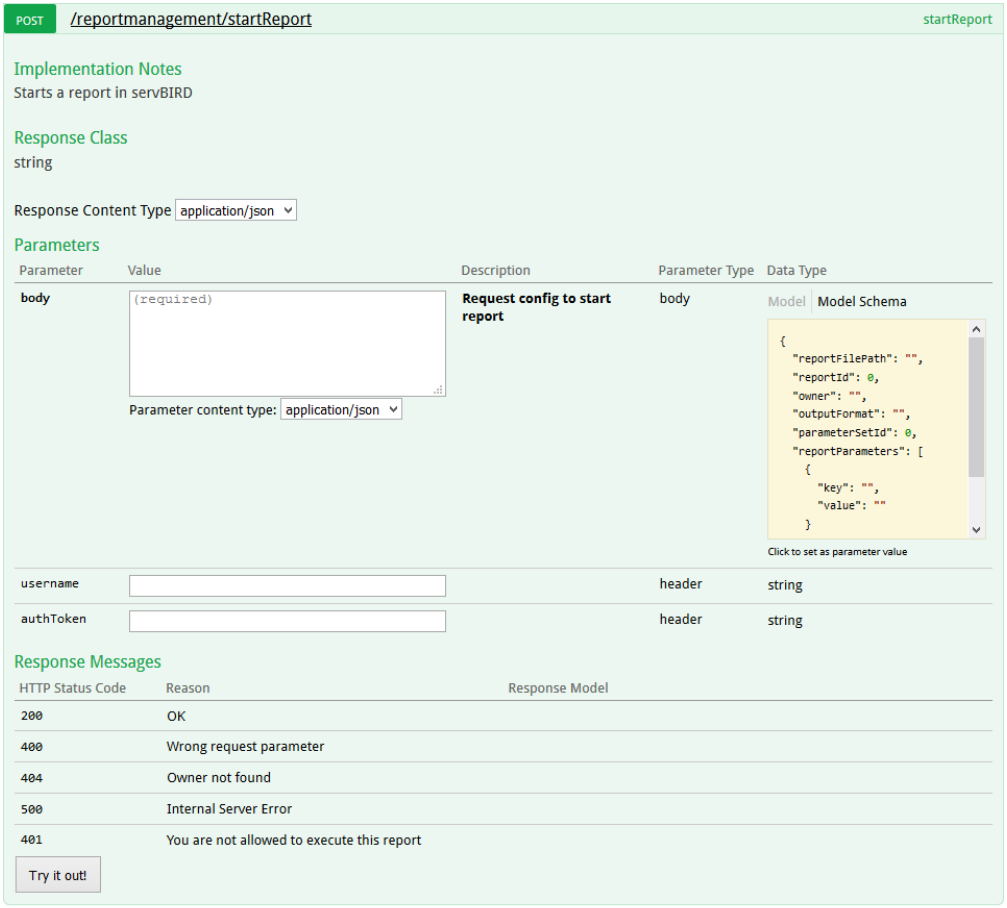
getReportParameterById

Alle Parameter zu einem Bericht (Parameter Berichtsdesign) über die Berichts Id



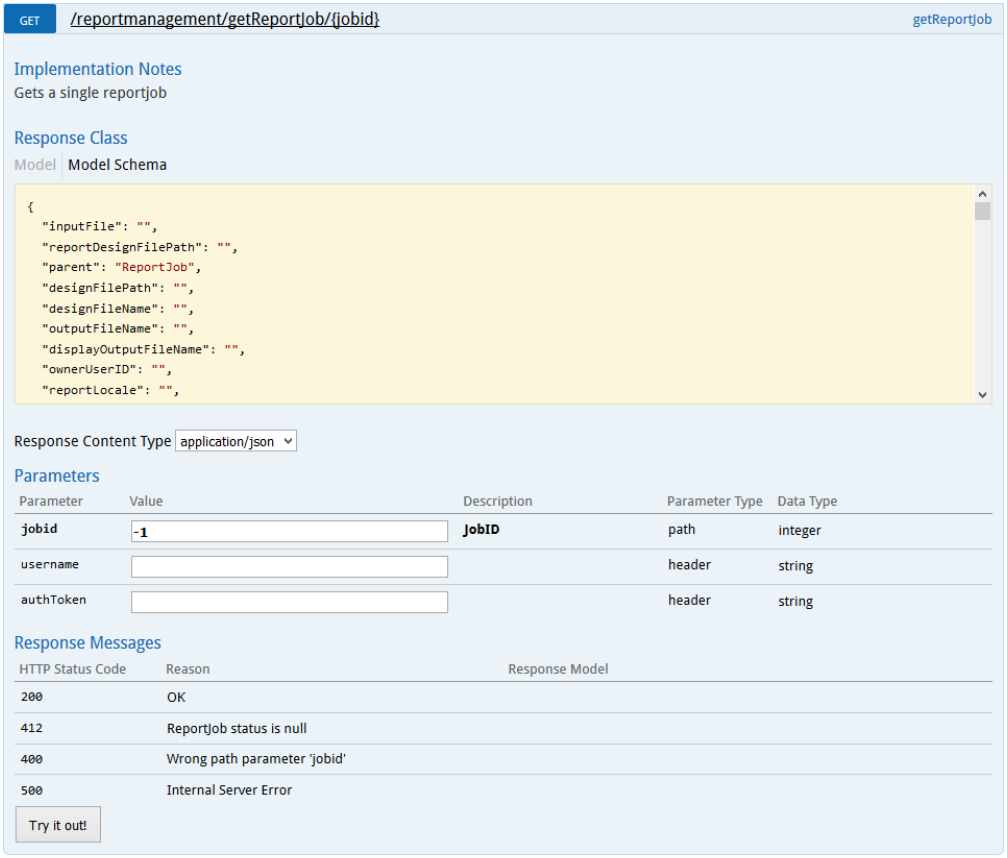
startReport

Bericht starten



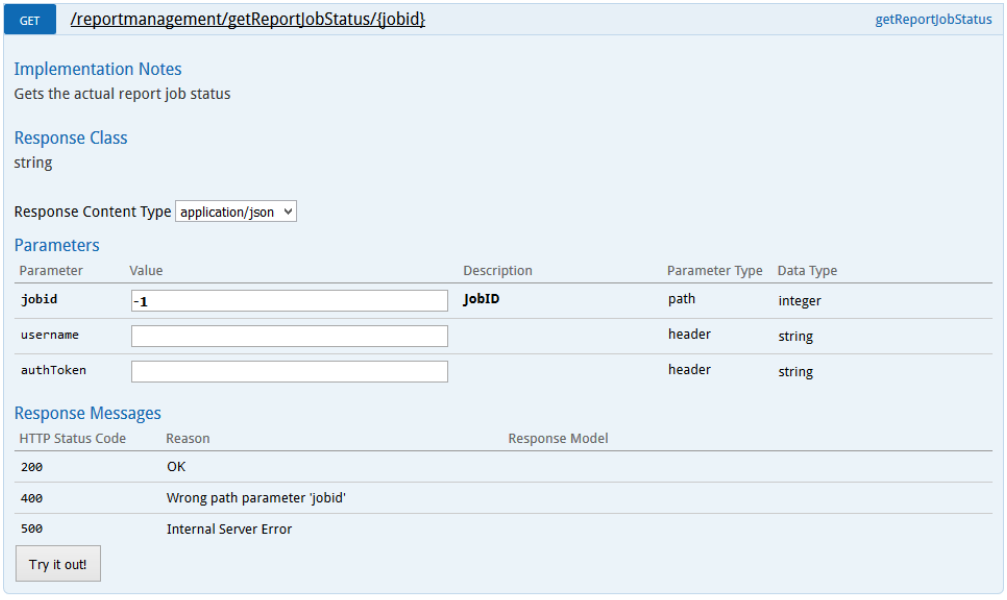
getReportJob

ReportJob Objekt abfragen



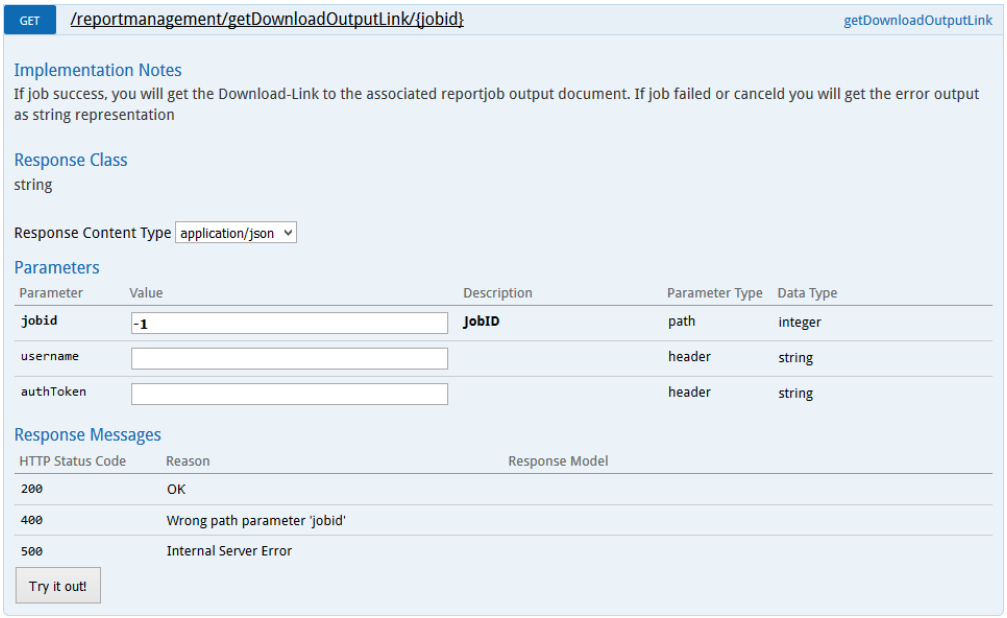
getReportJobStatus

Berichts Generierungsstatus Running, Success oder Failed



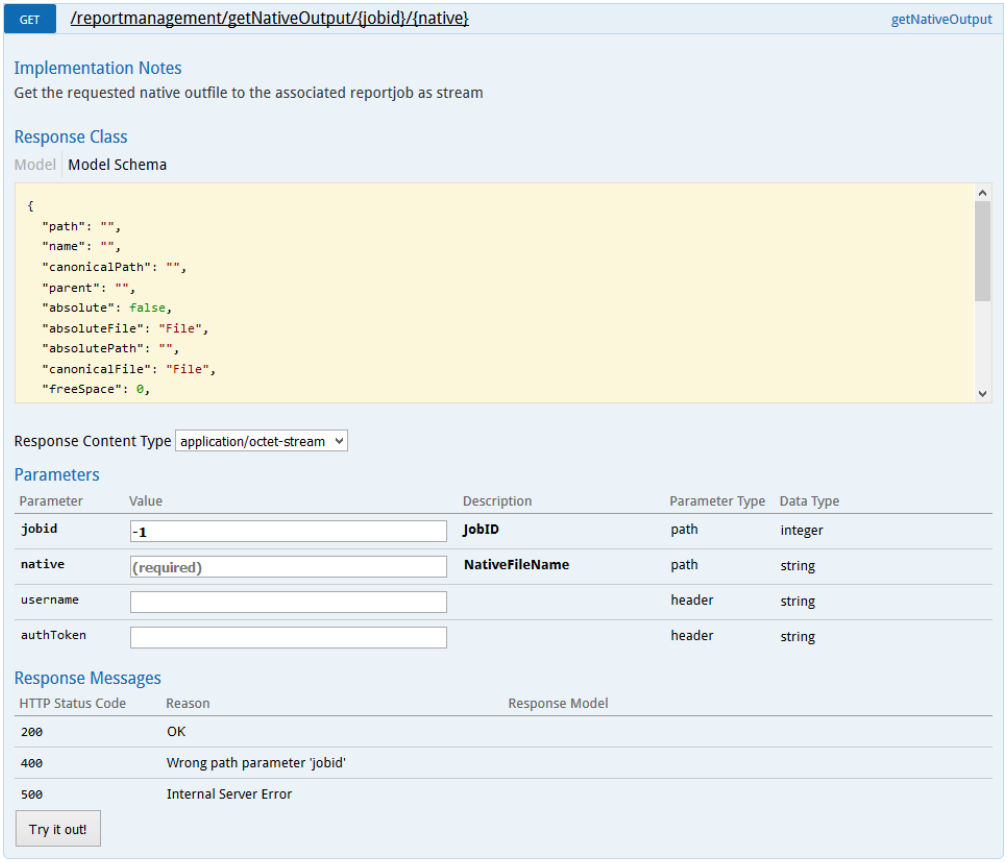
getDownloadOutputLink

Link auf das Ausgabeformat (HTML, PDF, etc.), wenn der Bericht erfolgreich generiert wurde



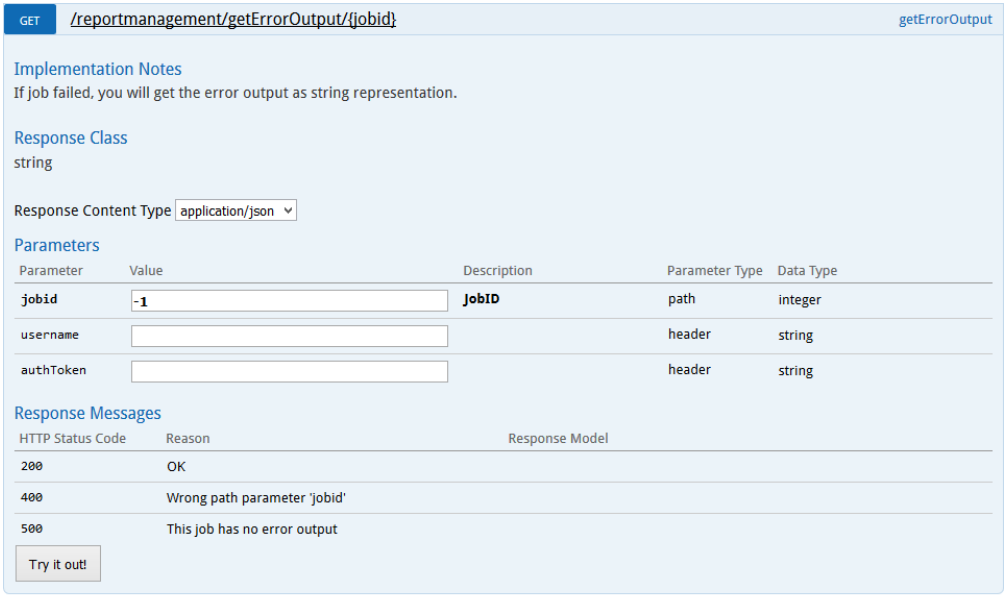
getNativeOutput

Byte-Stream Ausgabeformat (native Excel), wenn der Bericht erfolgreich generiert wurde.



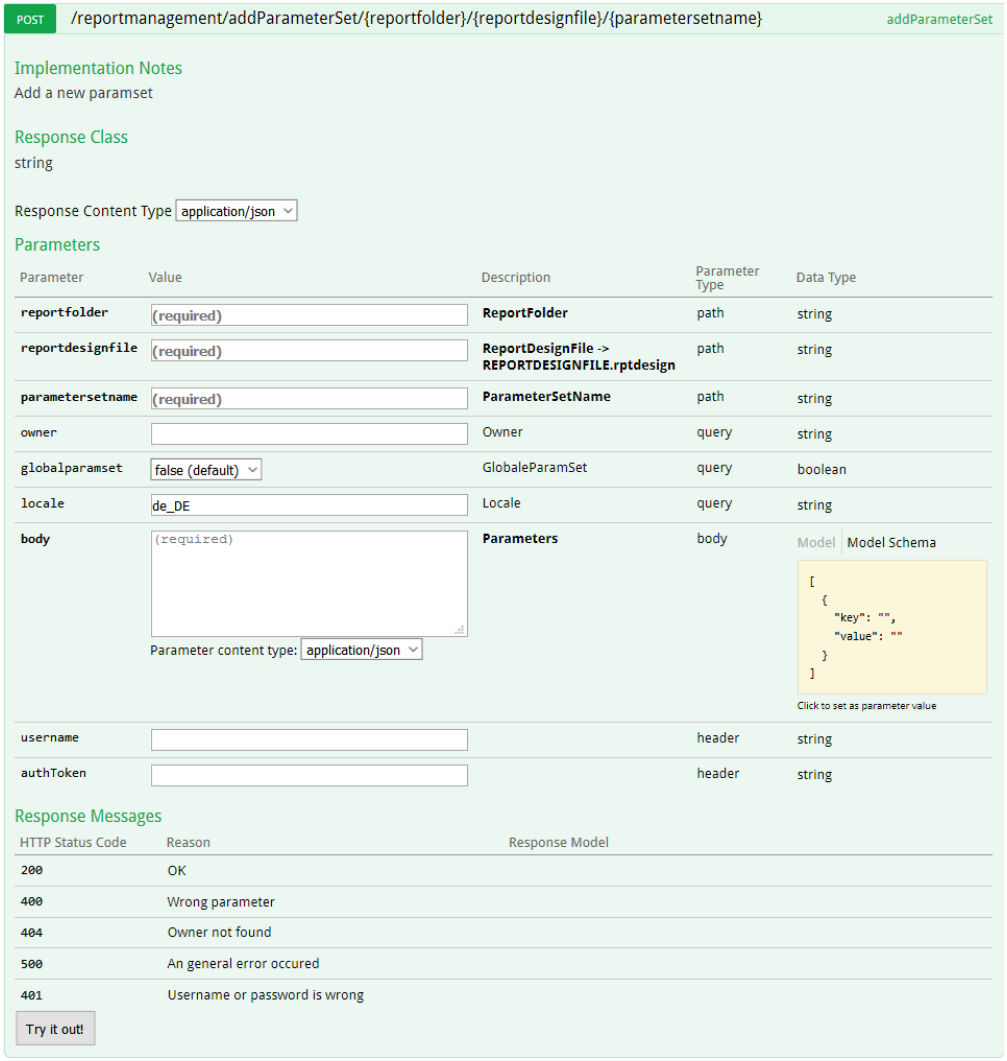
getErrorOutput

Ausgabe der Fehlermeldung, falls die Berichtsgenerierung fehlschlug.



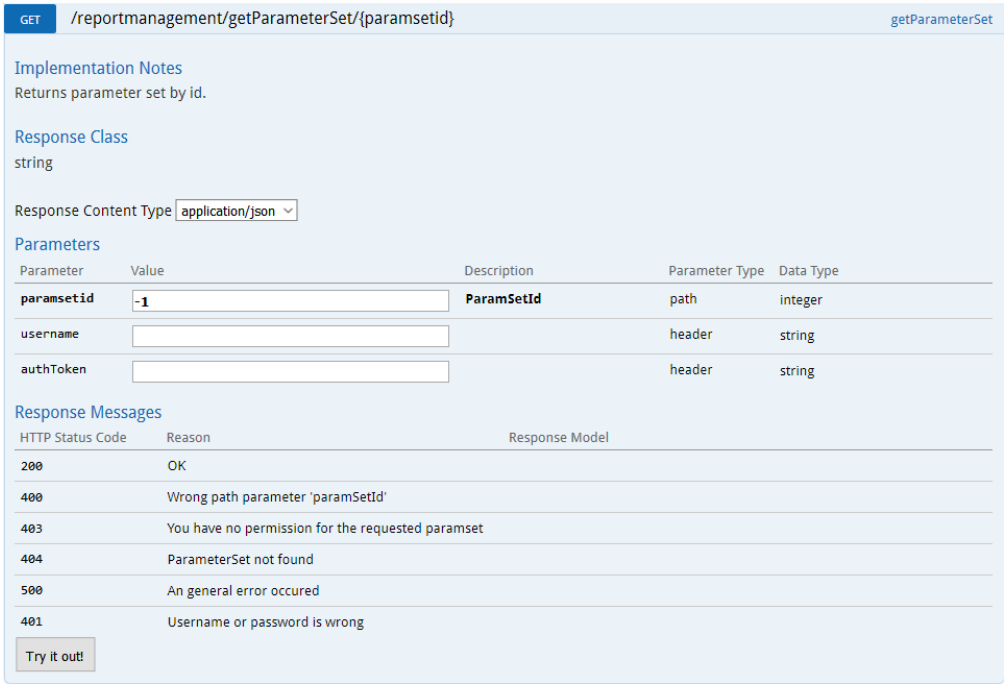
addParameterSet

Für einen Bericht ein neues Parameter Set anlegen



getParameterSet (ID)

Parameter Set ausgeben über die Parameter Set Id.



GET /reportmanagement/getParameterSet/{paramsetid} getParameterSet

Implementation Notes
Returns parameter set by id.

Response Class
string

Response Content Type application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
paramsetid	-1	ParamSetId	path	integer
username			header	string
authToken			header	string

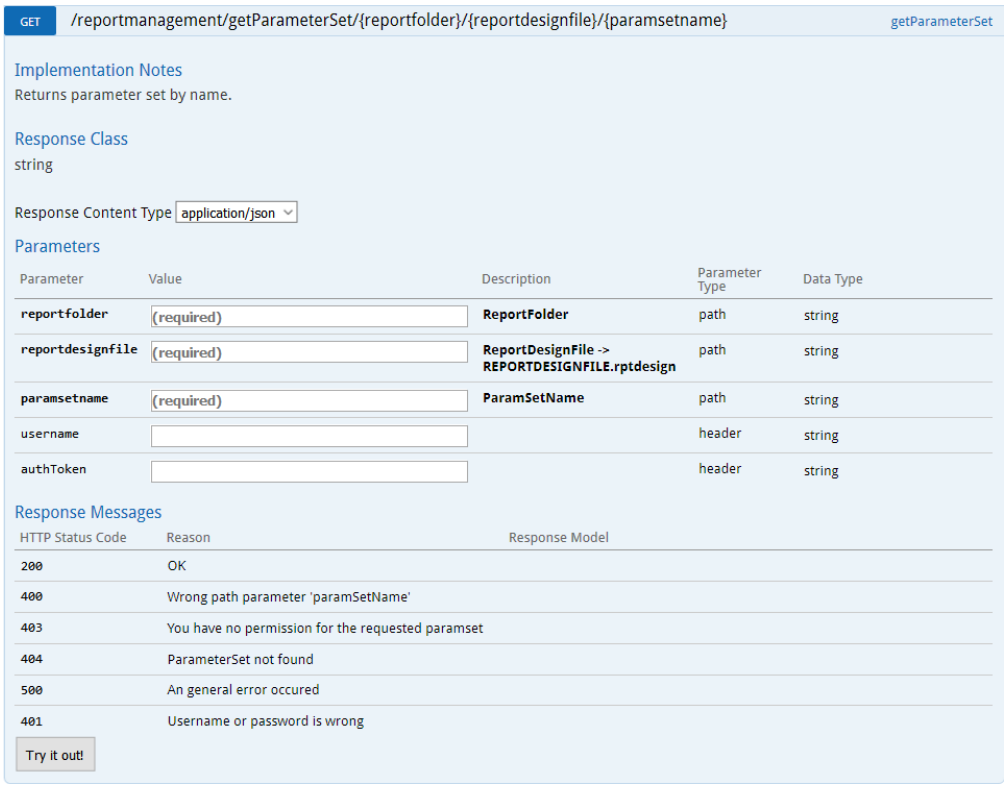
Response Messages

HTTP Status Code	Reason	Response Model
200	OK	
400	Wrong path parameter 'paramSetId'	
403	You have no permission for the requested paramset	
404	ParameterSet not found	
500	An general error occured	
401	Username or password is wrong	

Try it out!

getParameterSet (Bericht)

Alle Parameter Sets eines Berichts ausgeben.



GET /reportmanagement/getParameterSet/{reportfolder}/{reportdesignfile}/{paramsetname} getParameterSet

Implementation Notes
Returns parameter set by name.

Response Class
string

Response Content Type application/json

Parameters

Parameter	Value	Description	Parameter Type	Data Type
reportfolder	(required)	ReportFolder	path	string
reportdesignfile	(required)	ReportDesignFile -> REPORTDESIGNFILE.rptdesign	path	string
paramsetname	(required)	ParamSetName	path	string
username			header	string
authToken			header	string

Response Messages

HTTP Status Code	Reason	Response Model
200	OK	
400	Wrong path parameter 'paramSetName'	
403	You have no permission for the requested paramset	
404	ParameterSet not found	
500	An general error occured	
401	Username or password is wrong	

Try it out!

3.1.2 Installation und Konfiguration im Wildfly Applikation Server

Um mooBIRD auf dem Wildfly Applikation Server zu deployen ist es notwendig folgende Zeile in der Datei *"standalone.conf"* (Linux) bzw. *"standalone.conf.bat"* (Windows) einzutragen:

```
JAVA_OPTS="$JAVA_OPTS
-Dcom.sun.jersey.server.impl.cdi.lookupExtensionInBeanManager=true"
```

Dazu muss mooBIRD.war im Deployment-Verzeichnis bereitgestellt werden.

3.1.3 Hinweise zur Integration

In den folgenden Abschnitten wird auf einige Punkte hingewiesen, die bei der Verwendung der Funktionen der REST-Schnittstelle zu beachten sind.

Die detaillierten Angaben zu den Funktionen und Parametern entnehmen sie bitte dem vorangegangenen Abschnitt.

3.1.4 Benutzer an- und abmelden

Zunächst ist es notwendig einen Benutzer zu authentifizieren. Rufen Sie dazu die Methode `login` auf. Bei erfolgreicher Authentifizierung wird ein sogenannter `AuthToken` zurückgegeben.

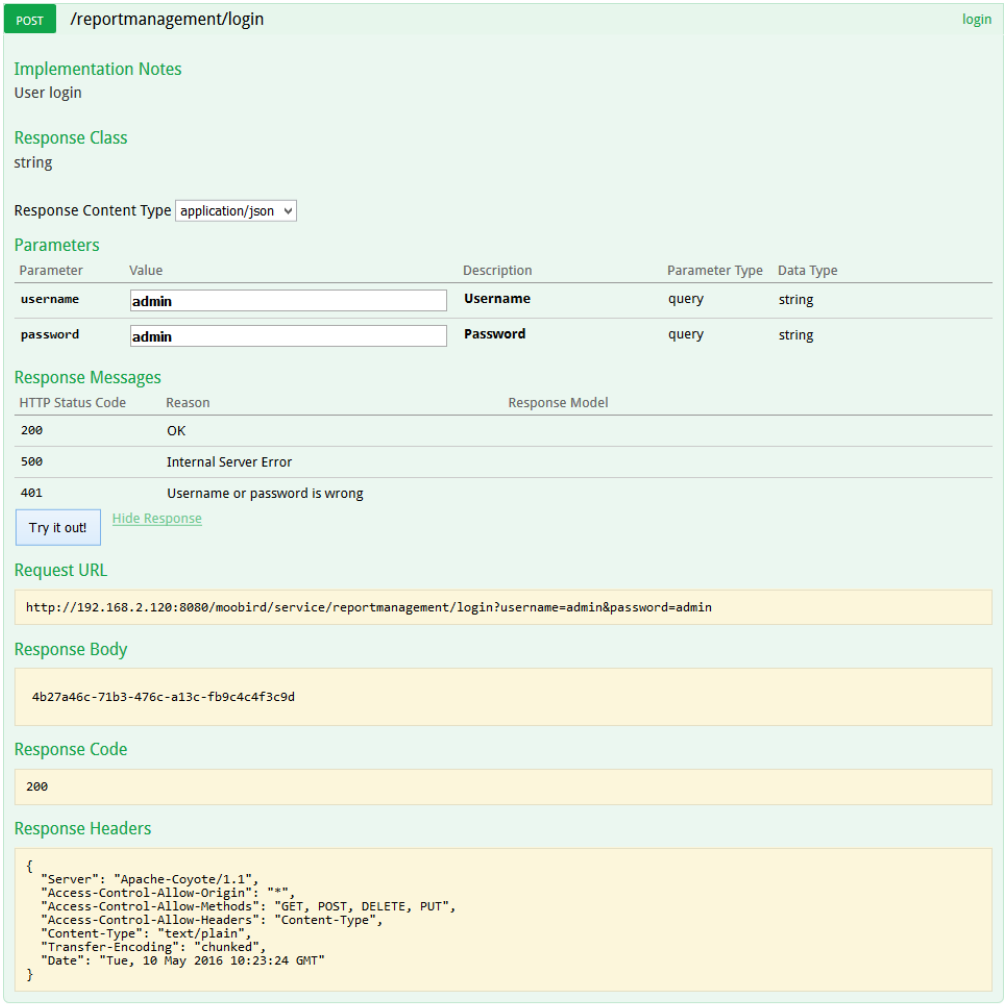
Ohne gültiges Token können vom Benutzer keine weiteren mooBIRD Funktionen ausgeführt werden. Das generierte Token, sowie der Benutzername müssen in sämtlichen Funktionsaufrufen, als Header-Parameter übergeben werden, ansonsten erhalten Sie eine Fehlermeldung.

Nach dem Aufruf der `logout` Funktion wird das Token für den Benutzer ungültig.

Ein weiterer Aufruf der Funktion `login` bewirkt, dass für diesen Benutzer ein neues Token generiert wird:

3.1.5 Bericht ausführen und Ergebnis abrufen

Um einen Bericht auszuführen und das Ausgabedokument abzurufen, sind folgende Schritte notwendig:



Beispiel Aufruf der Login Methode

1. Funktion: `startReport`
   (diese Funktion erwartet ein JSON-Objekt als Parameter, siehe Abschnitt "Funktionen")
2. Ergebnis: `ReportJobID`
3. Funktion: `getReportJobStatus(ReportJobId)`
    a. Ergebnis: `Running`
        i. Bericht wird noch ausgeführt
    b. Ergebnis: `Success`
        i. Funktion: `getDownloadOutputLink(ReportJobId)`
        ii. Ergebnis: Link auf die Ausgabedatei
    c. Ergebnis: `Failed`
        i. Funktion: `getErrorOutput(ReportJobID)`
        ii. Ergebnis: Fehlermeldung, gemäß dem TRADUI Toolbox Logging, im HTML Format

3.2 SOAP Webservice

Über einen Webservice (SOAP) kann **servBIRD** Fachanwendungen mit Berichten versorgen. Die Implementierung des Aufrufs der Webservice Schnittstelle muss in der Fachanwendung erfolgen. Die Schnittstelle unterstützt das Ausführen, Parametrisieren und Abrufen der Berichte über **servBIRD**.

servBIRD unterstützt, aus kompatibilitätsgründen, zwei Varianten der Implementierung des Webservices:

- Ohne optimierte Übertragung binärer Daten
- Mit optimierter Übertragung binärer Daten

Die Erfahrung hat gezeigt dass bestimmte Applikationen nur die "nicht optimierte" Variante unterstützen. Der Unterschied besteht auch im Funktionsumfang. Die "optmierte" Variante unterstützt auch das Streamen von Ausgabedokumenten, während die andere Variante nur Links zum Download zur Verfügung stellen kann.

Die WSDL ersterer Variante können Sie sich unter folgender Url anschauen:

http://HOST:PORT/BirdWebService/BirdWebServiceClassic/BirdWebServiceClassicBean?wsdl

Die zweite Variante unter:

http://HOST:PORT/BirdWebService/BirdWebService/BirdWebServiceBean?wsdl

Die vollständige Funktionsreferenz können Sie mit dieser HTML Datei im Anhang herunterladen:

4 Portal Integration

Für die Integration von **servBIRD** in ein Unternehmensportal steht zu jedem Menüpunkt von **servBIRD** ein direkter View zur Verfügung. Diese Views arbeiten ohne die **servBIRD** Menüsteuerung und sind jeweils gesondert über deren URL abrufbar.
Folgende Seiten werden als eigene Haupt-Views zur Verfügung gestellt:

- Einstieg und Hilfe
- dashBIRD
- BIRD Applications
- Berichte abrufen
- Berichte planen
- Fertige Berichte
- Aufgaben verwalten
- Einstellungen

Des Weiteren existiert ein gesonderter Login-View, auf den bei Bedarf automatisch umgeleitet wird.

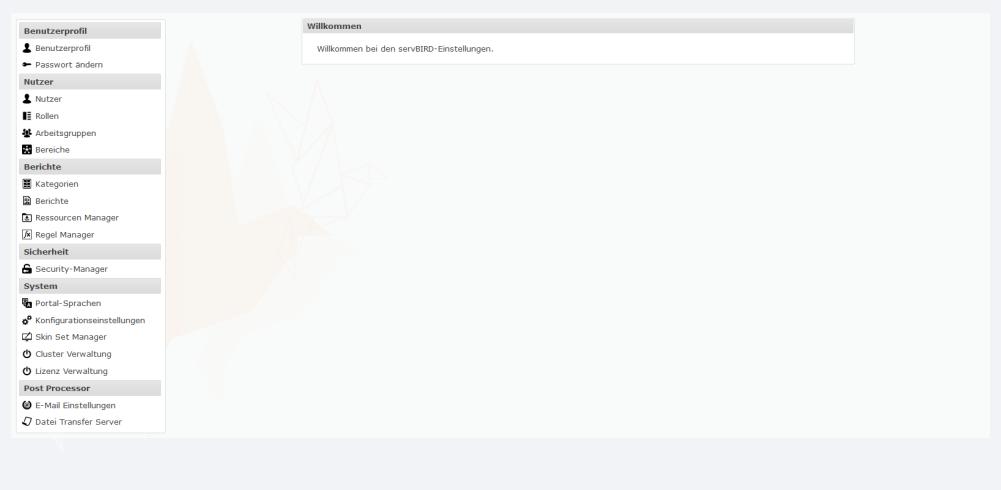
4.1 Vorschau Umgebung

Auf der Unterseite „`servBIRD/test/home.faces`“, kann man sich eine mögliche Integration der Views anschauen und selbst testen.

4.2 Views

4.2.1 Einstellungen

Link: servBIRD/admin/index.faces?intview=true



4.2.2 Dashbird

Link: servBIRD/portal/dashbird.faces?intview=true

Um ein Dashboard direkt zu generieren, muss die nachfolgende URL verwendet und zusätzliche URL Parameter verwendet werden (Der Parameter "dashboardid" ist für diese Funktionalität erforderlich. Der Parameter

"paramsetid" ist nur erforderlich wenn das Dashboard die Eingabe von Parameterwerten erfordert und der aufrufende Nutzer für dieses Dashboard kein Parameter Set als Favorit gesetzt hat.

Seite für direkte-Dashboard Ausführung: servBIRD/portal/dashbirdrun.faces

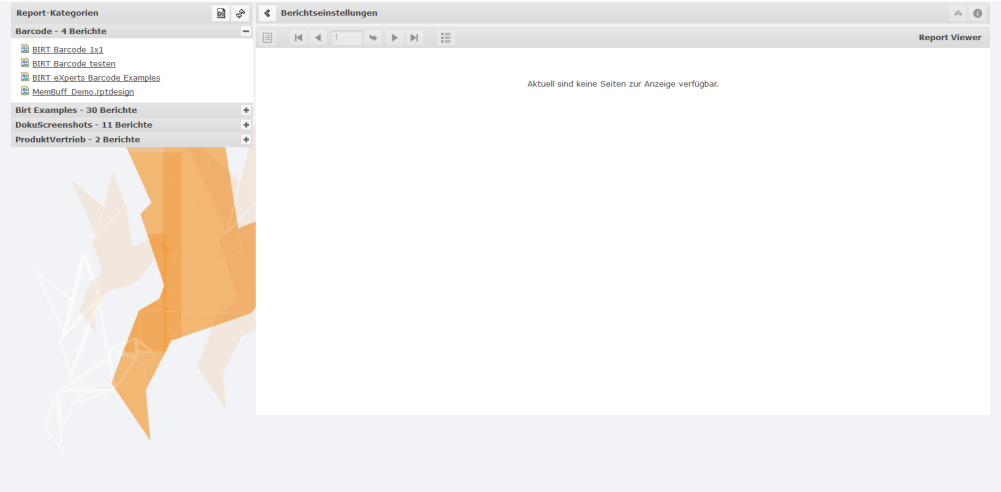
URL Parameter	Beschreibung	Beispiel
dashboardid	Die ID des aufzurufenden Dashboards	servBIRD/portal/dashbirdrun.faces?**dashboardid=42**
paramsetid	Die ID des zu verwendenen Parameter Sets	servBIRD/portal/dashbirdrun.faces?dashboardid=42&**paramsetid=128**

4.2.3 BIRD Applications

Link: servBIRD/portal/birdapplications.faces?intview=true

4.2.4 Berichte erstellen

Link: servBIRD/portal/runreport.faces?intview=true



Weitere URL Parameter

URL Parameter	Beschreibung	Syntax
run	Direktausführung eines Berichts	servBIRD/portal/runreport.faces?intview=true&**run=true**&reportfolder=/CATEGORY/&reportname=REPORTFILENAME.rptdesign

URL Parameter	Beschreibung	Syntax
reportfolder	Das Kategorieverzeichnis des Report Designs	servBIRD/portal/runreport.faces?intview=true&run=true&**reportfolder=/CATEGORY/**&reportname=REPORTFILENAME.rptdesign
reportname	Der Dateiname des Report Designs	servBIRD/portal/runreport.faces?intview=true&run=true&reportfolder=/CATEGORY/&**reportname=REPORTFILENAME.rptdesign**
outputFormat	Das Ausgabeformat Optional. Standardwert: "HTML"	servBIRD/portal/runreport.faces?intview=true&**outputFormat=PDF**&run=true&reportfolder=/CATEGORY/&reportname=REPORTFILENAME.rptdesign
showCategoryViewer	Steuert Anzeige des Reportbrowsers Optional. Standardwert: "true"	servBIRD/portal/runreport.faces?intview=true&run=true&**showCategoryViewer=false**&reportfolder=/CATEGORY/&reportname=REPORTFILENAME.rptdesign
--parametername	Übergabe von Parameterwerten	servBIRD/portal/runreport.faces?intview=true&run=true&showCategoryViewer=false&reportfolder=/CATEGORY/&reportname=REPORTFILENAME.rptdesign&**--param1=value1**&**--param2=value2**
multiValueDelimiter	Trennzeichen für Multivalue-Parameterwerte Optional. Standardwert: "|" (%7C)	servBIRD/portal/runreport.faces?intview=true&run=true&**multiValueDelimiter=%3B**&reportfolder=/CATEGORY/&reportname=REPORTFILENAME.rptdesign&**--param1=germany%3Bfrance%3Bsweden**

Hinweis

Parameterwerte müssen bei der Übergabe via URL entsprechend dem MIME Format "application/x-www-form-urlencoded" codiert sein.
Siehe hierzu: https://www.w3.org/TR/html4/interact/forms.html#h-17.13.4.1

Hinweis

Parameterwerte für Datumsparameter müssen in folgenden Formaten übergeben werden:

Date: yyyy-MM-dd

Time: HH:mm:ss

DateTime: yyyy-MM-dd HH:mm:ss

Information

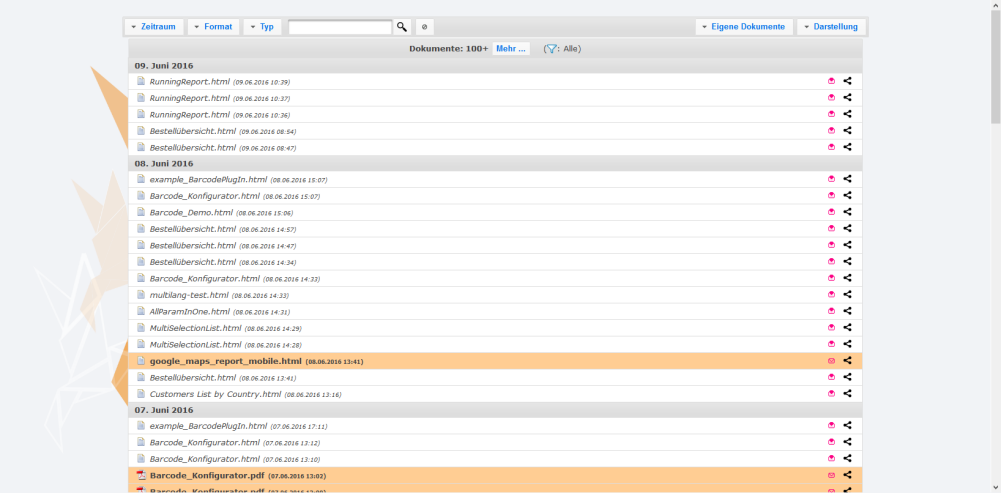
Multivalue-Parameterwerte können alternativ auch durch Mehrfachdefinition des Parameters in der URL angegeben werden. Beispiel:
servBIRD/portal/runreport.faces?intview=true&run=true&reportfolder=/CATEGORY/
&reportname=REPORTFILENAME.rptdesign&**--param1=germany**&**--param1=france**&**--param1=sweden**

4.2.5 Berichte planen

Link: servBIRD/portal/planschedulereport.faces?intview=true

4.2.6 Fertige Berichte

Link: servBIRD/portal/showreportdocuments.faces?intview=true



4.2.7 Aufgaben verwalten

Link: servBIRD/portal/showcompletereportjobs.faces?intview=true

Enthält auch Verweise auf laufende & wartende Berichte, sowie auf Zeitpläne.

Hinweise zur Integration

Bis zur Inbetriebnahme der SSO Funktionalität wird man bei der ersten Anmeldung bzw. bei Ablauf der Session auf den Login-View umgeleitet.
Die Views sollten innerhalb eines <iFrame> oder über ein AJAX-Get in das Portal integriert werden. Hierbei ist es wichtig, dass bei jedem neuen Seitenaufruf der Frame explizit neu geladen wird.

Ein Beispiel für die Integration innerhalb eines <iFrame>:

iFrame Codebeispiel

```
<iFrame id="iFrame" src="localhost:8080/servBIRD/portal/intviews/
showreportjobs.faces" style="max-width: 1600px; margin: auto;" frameborder="0"
 width="100%" height="980px" scrolling="no" ></iFrame>
```

Die Hintergrundfarbe der Views muss (sofern gewünscht) manuell angepasst werden.
Dazu wechselt man in den Administrationsbereich auf **Skin Set Manager**, aktiviert das Skin vom Typ „System Portal Skin“ und ändert die folgenden beiden Styles dementsprechend:

In diesem Fall wäre der innere, sowie äußere Hintergrund „weiß“.