

Internet-Anwendungen entwickeln mit Flow-Dateien: Tutorial



HELP.BCFESITFLOW

Release 4.6C



Copyright

© Copyright 2001 SAP AG. Alle Rechte vorbehalten.

Weitergabe und Vervielfältigung dieser Publikation oder von Teilen daraus sind, zu welchem Zweck und in welcher Form auch immer, ohne die ausdrückliche schriftliche Genehmigung durch SAP AG nicht gestattet. In dieser Publikation enthaltene Informationen können ohne vorherige Ankündigung geändert werden.

Die von SAP AG oder deren Vertriebsfirmen angebotenen Software-Produkte können Software-Komponenten auch anderer Software-Hersteller enthalten.

Microsoft[®], WINDOWS[®], NT[®], EXCEL[®], Word[®], PowerPoint[®] und SQL Server[®] sind eingetragene Marken der Microsoft Corporation.

IBM[®], DB2[®], OS/2[®], DB2/6000[®], Parallel Sysplex[®], MVS/ESA[®], RS/6000[®], AIX[®], S/390[®], AS/400[®], OS/390[®] und OS/400[®] sind eingetragene Marken der IBM Corporation.

ORACLE[®] ist eine eingetragene Marke der ORACLE Corporation.

INFORMIX[®]-OnLine for SAP und Informix[®] Dynamic Server[™] sind eingetragene Marken der Informix Software Incorporated.

UNIX[®], X/Open[®], OSF/1[®] und Motif[®] sind eingetragene Marken der Open Group.

HTML, DHTML, XML, XHTML sind Marken oder eingetragene Marken des W3C[®], World Wide Web Consortium, Massachusetts Institute of Technology.

JAVA[®] ist eine eingetragene Marke der Sun Microsystems, Inc.

JAVASCRIPT[®] ist eine eingetragene Marke der Sun Microsystems, Inc., verwendet unter der Lizenz der von Netscape entwickelten und implementierten Technologie.

SAP, SAP Logo, R/2, RIVA, R/3, ABAP, SAP ArchiveLink, SAP Business Workflow, WebFlow, SAP EarlyWatch, BAPI, SAPPHIRE, Management Cockpit, mySAP.com Logo und mySAP.com sind Marken oder eingetragene Marken der SAP AG in Deutschland und vielen anderen Ländern weltweit. Alle anderen Produkte sind Marken oder eingetragene Marken der jeweiligen Firmen.

Symbole

Symbol	Bedeutung
	Achtung
	Beispiel
	Empfehlung
	Hinweis
	Syntax
	Tip

Inhalt

Internet-Anwendungen entwickeln mit Flow-Dateien: Tutorial	5
Tutorial – Flow-Datei-Anwendungen: Voraussetzungen.....	9
Tutorial - Flow-Datei-Anwendungen: Beispielanwendung.....	10
Flow-Datei-Anwendung entwickeln: Wichtigste Schritte	12
Flow-Datei-Anwendung entwerfen.....	14
Geschäftslogik im R/3 definieren.....	21
SAP@Web Studio einrichten	23
SAP@Web Studio starten	24
ITS-Projekt anlegen.....	25
ITS-Site definieren.....	26
ITS-Servicedatei anlegen	28
Präsentation definieren	30
Flow-Logik definieren	32
Andere ITS-Dateien anlegen	35
ITS-Service publizieren	36
Dateien in die ITS-Source-Control laden	38
Flow-Datei-Anwendungen debuggen	40
Anhang A: Funktionsbausteine in der Beispielanwendung	41
Anhang B: ITS-Dateien in der Beispielanwendung	43

Internet-Anwendungen entwickeln mit Flow-Dateien: Tutorial

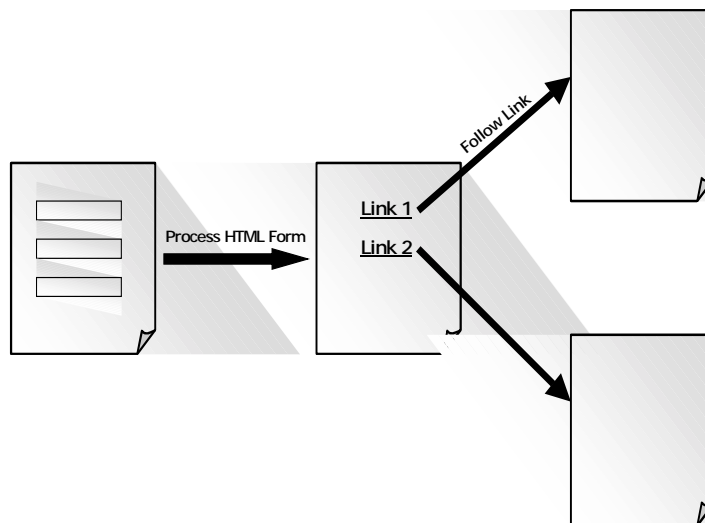
Einsatzmöglichkeiten

Diese Dokumentation ist ein Tutorial mit einer Schritt-für-Schritt-Anleitung, wie Sie Internet-Anwendungen entwickeln, die über den Internet Transaction Server (ITS) laufen und Flow-Dateien verwenden.

Mit dem Modell zur Flow-Datei-Implementierung können Sie Internet-Anwendungen entwickeln, die aus verbundenen HTML-Seiten bestehen. Diese Seiten können Sie mit Daten aus dem R/3-System (oder jedem anderen externen System) füllen. Die Seiten bieten eine Reihe von Anwendungsfunktionen; sie werden mittels Verfolgen von Hyperlinks oder Verarbeiten von HTML-Formularen erzeugt. Der Dialogablauf wird auf Client-Seite vom Benutzer bestimmt, der frei zwischen den Seiten navigieren kann.

Da der Dialogablauf nicht vorab festgelegt wird, hängt viel davon ab, was der Benutzer tut. Dies steht im Gegensatz zum Dialogablauf in anderen Geschäftsszenarien, in denen die Geschäftsanwendung die Navigationsmöglichkeiten des Benutzers einschränken kann.

Die nachfolgende Grafik verdeutlicht das Grundkonzept:



Dieses Tutorial beschreibt die Entwicklung von Anwendungen, die Flow-Dateien verwenden und deren Geschäftslogik in remote-fähigen Funktionsbausteinen (RFCs)

Internet-Anwendungen entwickeln mit Flow-Dateien: Tutorial

implementiert ist, die aus dem R/3-System aufgerufen werden. Sie können die Geschäftslogik auch mit Business APIs (BAPIs) implementieren.

In zukünftigen Releases wird es möglich sein, Bausteinaufrufe aus jedem beliebigen externen System zu definieren.

Einführungshinweise

Dieses Implementierungsmodell sollten Sie für solche Anwendungen in Betracht ziehen, die viele Anwendungsfunktionen auf einer Seite anbieten, und bei denen der Dialogablauf nicht von vornherein festgelegt ist.

Typische Flow-Datei-Anwendungen haben einfache Benutzungsoberflächen, auf denen der Benutzer viel anklicken kann, nur wenige Daten von Hand eingeben muß, und kaum Daten formatieren muß. Sie werden oft in Szenarien des elektronischen Handels (e-Commerce) eingesetzt.

Integration

Um Anwendungen zu entwickeln, die Flow-Dateien verwenden, bei denen die Geschäftslogik in Bausteinaufrufen aus dem R/3-System implementiert ist, müssen Sie folgende Komponenten installieren:

- Den ITS
Der ITS bildet die Schnittstelle zwischen dem R/3-System und dem Internet.
- Das SAP@Web Studio
Das SAP@Web Studio ist ein PC-Werkzeug zur Implementierung von Services, die alle vom ITS zur Ausführung von Anwendungen benötigten Dateien enthalten.
- Das R/3-System

Funktionsumfang

Wie alle anderen Implementierungsmodelle zur Entwicklung von ITS-gesteuerten Internet-Anwendungen ermöglicht Ihnen auch dieses Modell die Entwicklung von Anwendungen, die Dokumente im HTML-Format zurück an den Web-Browser-Client geben, da dieses Format von allen größeren Web-Browsern verarbeitet werden kann.

Wie bei allen anderen Implementierungsmodellen gibt es auch hier eine klare Trennung zwischen Geschäftslogik und Präsentationsaspekten. In diesem Fall ist die Definition des Dialogablaufes auch eine separate Aufgabe.

- Sie implementieren einen Satz Bausteine, die die Geschäftslogik im R/3-System (oder einem anderen externen System) enthalten.
Wenn Sie die Geschäftslogik im R/3-System implementieren, legen Sie Business APIs (BAPIs) oder remote-fähige Standardfunktionsbausteine mit dem Function Builder in der ABAP Workbench an. Dieses Tutorial verwendet remote-fähige Funktionsbausteine .
- Sie implementieren die Präsentation und den Dialogablauf im SAP@Web Studio.
Dazu müssen Sie einen ITS-Service anlegen, der alle zur Implementierung und Ausführung der Anwendung benötigten Dateien enthält.
 - Die Präsentation legt das äußere Erscheinungsbild jeder Anwendung fest.

Internet-Anwendungen entwickeln mit Flow-Dateien: Tutorial

Sie entwerfen die Präsentation, indem Sie einen Satz HTML^{Business}-Templates anlegen.

- Der Dialogablauf legt fest, welches Template wann angezeigt wird, abhängig von der gewählten Aktion des Benutzers.

Sie implementieren den Dialogablauf, indem Sie die Logik in Flow-Dateien definieren. Diese Flow-Logik definiert die Reihenfolge, in der Bausteine aufgerufen werden, auf der Grundlage von Ereignissen und Ausnahmen.

Es gibt eine Flow-Datei für jedes HTML^{Business}-Template, das eine Dialogablaufdefinition benötigt.

Nachdem Sie die Geschäftslogik in R/3 definiert und alle von ITS zur Ausführung einer Anwendung benötigten Dateien im SAP@Web Studio angelegt haben, können Sie die ITS-Dateien in die Source-Control im R/3 hochladen und die Anwendung testen. Benutzer können die Anwendung von jedem geeigneten Web-Browser aus starten.

Das Tutorial enthält:

Dieses Tutorial sollten Sie der Reihe nach von Anfang bis Ende durchlesen. Es enthält alle Schritte, die Sie ausführen müssen, wenn Sie Internet-Anwendungen mit Flow-Dateien entwickeln wollen. Es bezieht sich dabei auf eine bestimmte Beispielanwendung.

Weitere Informationen

Eine vollständige Beschreibung des Implementierungsmodells sowie eine Liste der Syntaxelemente der Flow-Logik und eine Beschreibung ihrer Verwendung finden Sie unter:

[Internet-Anwendungen entwickeln mit Flow-Dateien: Referenzhandbuch \[Extern\]](#)

Weitere Informationen zu anderen ITS-Implementierungsmodellen sowie weiteren Aspekten des ITS finden Sie in folgenden Dokumenten:

[ITS-Implementierungsmodelle \[Extern\]](#)

[Entwicklung von Internet-Anwendungen \[Extern\]](#)

[SAP@Web Studio \[Extern\]](#)

[HTMLBusiness-Sprachbeschreibung \[Extern\]](#)

[ITS-Verwaltung \[Extern\]](#)

Wenn diese Links von Ihrem aktuellen Arbeitsplatz aus nicht funktionieren, finden Sie diese Dokumentationen:

- Im R/3-System:
Wählen Sie *Hilfe* → *SAP-Bibliothek* → *Basis* → *Frontend Services (BC-FES)* → *ITS / SAP@Web Studio (BC-FES-ITS)*
- Im SAP@Web Studio:
Wählen Sie *Help* .

Das Tutorial enthält nicht:

Dieses Tutorial enthält **keine** Beschreibung, wie Sie remote-fähige Funktionsbausteine oder BAPIs anlegen. Es geht davon aus, daß Sie mit den ABAP-Programmiertechniken vertraut sind

Internet-Anwendungen entwickeln mit Flow-Dateien: Tutorial

und daß Sie wissen, wie Sie die diversen Programmierwerkzeuge der ABAP Workbench im R/3-System einsetzen.

Weitere Informationen

Weitere Informationen zu den ABAP-Programmiertechniken, ABAP Workbench-Werkzeugen und zur BAPI-Programmierung finden Sie unter den entsprechenden Links weiter unten.

Wenn diese Links von Ihrem aktuellen Arbeitsplatz aus nicht funktionieren, und Sie nicht im R/3-System sind, melden Sie sich im neusten Release an und folgen Sie den Anweisungen für die Dokumentation, die Sie anzeigen wollen.

- [ABAP Programmierung \[Extern\]](#)

Im R/3-System wählen Sie *Hilfe* → *SAP-Bibliothek* → *Basis* → *ABAP-Programmierung und Laufzeitumgebung (BC-ABA)* → *BC – ABAP Programmierung*

- [ABAP Workbench: Werkzeuge \[Extern\]](#)

Im R/3-System wählen Sie *Hilfe* → *SAP-Bibliothek* → *Basis* → *ABAP Workbench (BC-DWB)* → *BC – ABAP Workbench: Werkzeuge*

- [ABAP Workbench Tutorial \[Extern\]](#)

Im R/3-System wählen Sie *Hilfe* → *SAP-Bibliothek* → *Basis* → *ABAP Workbench (BC-DWB)* → *BC – ABAP Workbench Tutorial*

- [BAPI Programmierung \[Extern\]](#)

Im R/3-System wählen Sie *Hilfe* → *SAP-Bibliothek* → *Anwendungsübergreifende Komponenten* → *Business Framework Architecture (CA-BFA)*

Tutorial – Flow-Datei-Anwendungen: Voraussetzungen

Ehe Sie anfangen, dieses Tutorial durcharbeiten, sollten Sie:

- Den Internet Transaction Server (ITS) Release 4.6C (oder höher) installieren.
- Das SAP@Web Studio Release 4.6C (oder höher) installieren.
- Das R/3-System Release 4.6C (oder höher) installieren.

Um die Funktionsbausteine und ITS-Objekte anzuzeigen, aus denen die Beispielanwendung besteht, benötigen Sie Release 4.6C oder höher.

Um Anwendungen mit Flow-Dateien zu entwickeln, benötigen Sie Release 4.6A oder höher.



Einige Business APIs (BAPIs) sind in R/3-Releases vor 4.6B nicht verfügbar. Daher empfiehlt SAP, R/3-Release 4.6B oder höher zu verwenden, wenn Sie vorhaben, Flow-Datei-Anwendungen zu entwickeln, die die Geschäftslogik über BAPIs definieren.

In Zukunft wird es möglich sein, Anwendungen zu entwickeln, die Flow-Dateien in früheren R/3-Releases verwenden.

Zusätzlich sollten Sie:

- Sicherstellen, daß die remote-fähigen Funktionsbausteine und alle erforderlichen ITS-Dateien, die die Beispielanwendung bilden, im R/3-System verfügbar sind.
 - Die Entwicklungsklasse ist COM_ITS_EXAMPLES.
 - Funktionsgruppe AUCTION_FUNCTIONS
 - Diese Funktionsgruppe enthält die Funktionsbausteine, die zur Implementierung der Geschäftslogik verwendet werden.
 - Internet-Service AUCTIONNEW
 - Dieser ITS-Service enthält die ITS-Dateien – Servicedateibeschreibung, HTML^{Business}-Templates und Flow-Dateien –, die zur Implementierung der Präsentation und des Dialogablaufs verwendet werden.
- Debugger-Funktionen einrichten.
 - Mit dem ITS-Debugger können Sie eine Web-Transaktion von Ihrem Web-Browser aus starten und der Verarbeitung auf der R/3-Seite in einem SAPGUI-Fenster folgen.
 - Weitere Informationen finden Sie unter [Debuggen von Flow-Datei-Anwendungen \[Seite 40\]](#).

Wenn Ihr Unternehmen einen ITS-Administrator hat, sind die meisten dieser Aufgaben wahrscheinlich bereits ausgeführt.

Weitere Informationen finden Sie im SAPNet-Hinweis 191373.

Tutorial - Flow-Datei-Anwendungen: Beispielanwendung

Tutorial - Flow-Datei-Anwendungen: Beispielanwendung

Dieses Tutorial zeigt anhand einer Beispielanwendung die Schritte, die Sie durchführen müssen, um eine Internet-Anwendung mit Flow-Dateien zu entwickeln.

Die Beispielanwendung ist ein integriertes Auktionsszenarium, das registrierten Benutzern ein Forum bietet, in dem Sie zum Verkauf stehende Objekte anschauen, selbst Objekte verkaufen und für Objekte Gebote abgeben können. Sie enthält außerdem Verwaltungs- und Suchfunktionen.

Die Beispielanwendung besteht aus folgenden Aspekten:

- Geschäftslogik
- Service-Datei, Präsentation und Dialogablauf

Geschäftslogik

Die Geschäftslogik ist definiert in einem Satz remote-fähiger Standardfunktionsbausteine, die mit Ihrem R/3-System ausgeliefert werden.

Eine Liste der in dieser Anwendung verwendeten Funktionsbausteine finden Sie in [Anhang A: Funktionsbausteine in der Beispielanwendung \[Seite 41\]](#) am Ende dieses Tutorials.



Die Beispielanwendung in diesem Tutorial verwendet remote-fähige Standardfunktionsbausteine; Sie können jedoch ebenso Business APIs (BAPIs) verwenden.

Service-Datei, Präsentation und Dialogablauf

Damit Sie die Anwendung in einem Web-Browser ausführen können, enthält die Anwendung auch folgende ITS-Dateien.

- Service-Datei
Die Service-Datei enthält alle Parameter, die der ITS zur Ausführung der Anwendung benötigt.
- HTML^{Business}-Templates
HTML^{Business}-Templates definieren die Präsentation oder das äußere Erscheinungsbild ("Look&Feel") der Anwendung.
Es gibt ein Template für jedes Anwendungsdynpro.
- Flow-Dateien
Flow-Dateien enthalten die Logik, die den Dialogablauf der Anwendung definiert (Flow-Logik).
Jedem HTML^{Business}-Template, das eine Dialogablaufdefinition benötigt, ist eine Flow-Datei zugeordnet.

Alle diese Dateien sollten in Ihrem R/3-System zur Verfügung stehen.

Tutorial - Flow-Datei-Anwendungen: Beispielanwendung

Sie können die HTML^{Business}-Templates und Flow-Dateien aus dem Beispiel als Grundlage für den Entwurf Ihrer eigenen Dateien verwenden.

Eine Liste aller in der Anwendung verwendeten ITS-Dateien – Servicedatei, HTML^{Business}-Templates und Flow-Dateien – finden Sie in [Anhang B: ITS-Dateien in der Beispielanwendung \[Seite 43\]](#) am Ende dieses Tutorials.

Beispielanwendung ausführen

Wenn Sie alle Funktionsbausteine und ITS-Dateien in Ihrem R/3-System haben, können Sie die Beispielanwendung von Ihrem Web-Browser aus ausführen, unter folgender URL:

`http://<myhost>.myport/scripts/wgate/auctionnew/!`

Sie sehen dann das erste Dynpro der Anwendung. Probieren Sie aus, wie es funktioniert.

Flow-Datei-Anwendung entwickeln: Wichtigste Schritte

Einsatzmöglichkeiten

Dieser Ablauf beschreibt die wichtigsten Schritte beim Entwurf einer Internet-Anwendung, die vom Internet Transaction Server (ITS) gesteuert wird und Flow-Dateien verwendet.

Voraussetzungen

Legen Sie fest, daß Sie Ihre Anwendung nach diesem Implementierungsmodell entwickeln wollen.

Ablauf

Um eine Anwendung anzulegen, die Flow-Dateien verwendet, müssen Sie:

1. Die [Anwendung entwerfen \[Seite 14\]](#).

Dazu müssen Sie ein grundlegendes Konzept dessen haben, was Ihre Anwendung tun soll. Dann entscheiden Sie, welche Templates Sie benötigen, wie und wann sie aufgerufen werden, und welche Geschäftslogik Sie definieren müssen, um die Daten zu verarbeiten.
2. Die [Geschäftslogik \[Seite 21\]](#) im R/3-System definieren.
3. Den Rest der Anwendung im SAP@Web Studio entwickeln.
 - a. Die [Umgebung einrichten \[Seite 23\]](#).
 - b. Einen [ITS-Service anlegen \[Seite 28\]](#) um die Anwendung auszuführen.

Jede Anwendung hat eine Servicebeschreibung. Sie enthält die Parameter, die definieren, wie die Anwendung ausgeführt wird.
 - c. Die [Präsentation definieren \[Seite 30\]](#).

Jede Anwendung besteht aus diversen verbundenen HTML-Seiten, die dem Benutzer Funktionen und Hyperlinks anbieten; daher müssen Sie einen Satz HTML^{Business}-Templates anlegen.

HTML^{Business}-Templates enthalten Standard-HTML-Anweisungen und HTML^{Business}-Anweisungen. HTML^{Business} ist eine SAP-spezifische Makrosprache, die es Ihnen ermöglicht, R/3-Daten zur Laufzeit in Templates einzumischen.
 - d. Die [Flow-Logik definieren \[Seite 32\]](#).

Zu jedem HTML^{Business}-Template, das eine Dialogablaufdefinition benötigt, gibt es eine zugeordnete Flow-Datei. Diese Flow-Datei definiert die Flow-Logik, die dem Benutzer die Navigation in der Anwendung ermöglicht.

Flow-Dateien enthalten XML-Anweisungen (eXtensible Markup Language) und verwenden ein [vordefiniertes Subset \[Extern\]](#) von XML-Elementen.
 - e. Alle anderen [ITS-Dateien anlegen \[Seite 35\]](#), die Sie benötigen.

Zu diesen anderen Dateien gehören Sprachressourcen und MIME-Dateien (Multipurpose Internet Mail Extension).

Flow-Datei-Anwendung entwickeln: Wichtigste Schritte

- Sprachressourcendateien enthalten alle in HTML^{Business}-Templates verwendeten sprachenspezifischen Texte. Diese Dateien sind optional. Allerdings ist es sinnvoll, sie zu verwenden, da Sie so mit einem einzigen Satz von HTML^{Business}-Templates alle Sprachen abdecken können.
 - MIME-Dateien enthalten alle Grafiken sowie Bild-, Ton- und Videokomponenten, die Sie in Ihren ITS-Service einbinden wollen. Diese Dateien sind auch optional.
- f. Den [Service an eine ITS-Site publizieren \[Seite 36\]](#).
 - g. Die [Dateien zur ITS-Source-Control hinzufügen \[Seite 38\]](#).
4. Ihre [Anwendung debuggen \[Seite 40\]](#).

Ergebnis

Nachdem Sie diese Schritte ausgeführt haben, sollten Sie eine vollständige Flow-Datei-Anwendung haben, die Sie von jedem geeigneten Web-Browser aus ausführen können.



Obwohl es für den Ablauf nicht wichtig ist, ob Sie zuerst die Geschäftslogik implementieren oder zuerst die Präsentation und den Dialogablauf definieren, ist es von der Arbeitsweise her sinnvoll, die in diesem Tutorial beschriebene Reihenfolge beizubehalten.

Flow-Datei-Anwendung entwerfen

Flow-Datei-Anwendung entwerfen

Verwendung

Wenn Sie sich entschließen, eine Internet-Anwendung mit Flow-Dateien zu implementieren, müssen Sie eine präzise Vorstellung von der Anwendung haben, die Sie entwickeln wollen.

Sie müssen folgendes wissen:

- Den Hauptzweck der Anwendung.
- Welche Funktionen die Anwendung dem Benutzer bieten soll.
- Wie der Benutzer navigieren können soll.
- Wie die Anwendung aussehen soll.

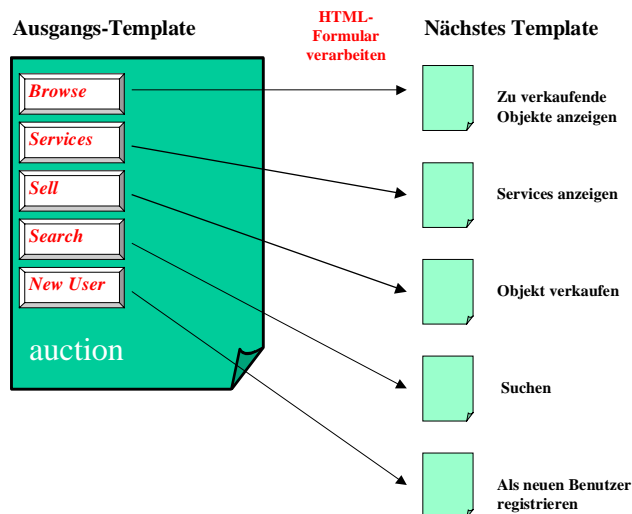
Nachdem Sie diese Fragen beantwortet haben, werden Sie wissen:

- Welche Funktionsbausteine Sie benötigen, um die Geschäftslogik im R/3 anzulegen.
- Welche Templates Sie benötigen und wie Sie sie organisieren sollten.
- In welcher Reihenfolge die Templates verarbeitet werden.

Vorgehensweise

Um Ihre Anwendung klar darzulegen, beginnen Sie mit einem Zustandsdiagramm.

Wenn Sie die Beispielanwendung AUCTION als Vorlage verwenden, können Sie sich den Ausgangszustand in etwa so vorstellen:



Flow-Datei-Anwendung entwerfen

Das Ausgangs-Template dieser Anwendung bietet diverse Funktionen. Wenn der Benutzer eine dieser Funktionen wählt, werden Sie ein oder mehrere Funktionsbausteine aufrufen und das entsprechende Template anzeigen müssen, oder Sie werden nur das nächste Template anzeigen müssen. Auf dem nächsten Template wiederholt sich dieser Ablauf.

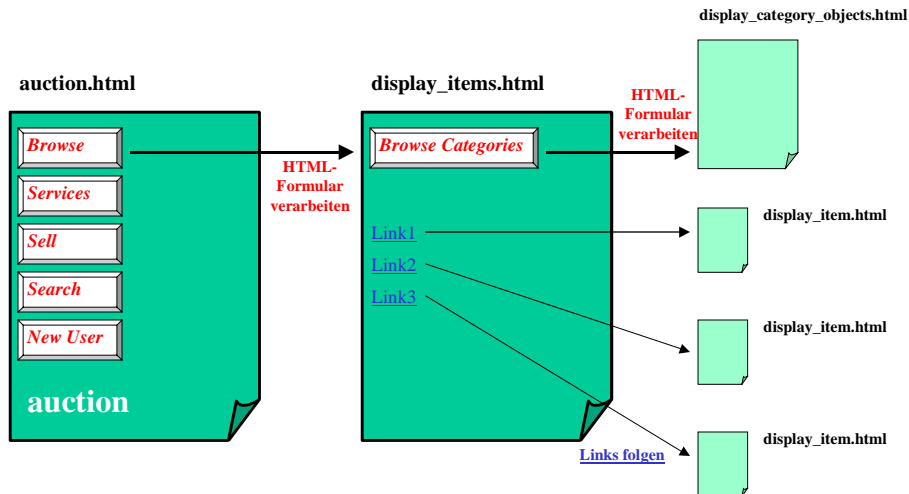
Wie in obiger Grafik gezeigt, kann das Ausgangs-Template `auction.html` die folgenden Funktionen auf fünf Drucktasten anbieten:

- *Browse*
- *Services*
- *Sell*
- *Search*
- *New User*

Jede dieser Funktionen könnte den Benutzer auf Templates führen, die weitere Eingabefelder, Drucktasten und Links enthalten.

Wir wollen nun die einzelnen Funktionen von `auction.html` etwas genauer betrachten und sehen, wie sich der Dialogablauf entwickeln könnte:

Browse



Wenn der Benutzer *Browse* wählt, könnte das nächste Template `display_items.html`:

- Eine Liste aller zum Verkauf stehenden Objekte in der Datenbank anzeigen.
 Jedes Objekt könnte einen Link zu Einzelheiten über das Objekt enthalten. Dies würde Informationen über den Verkäufer mit einschließen und es dem Benutzer ermöglichen,

Flow-Datei-Anwendung entwerfen

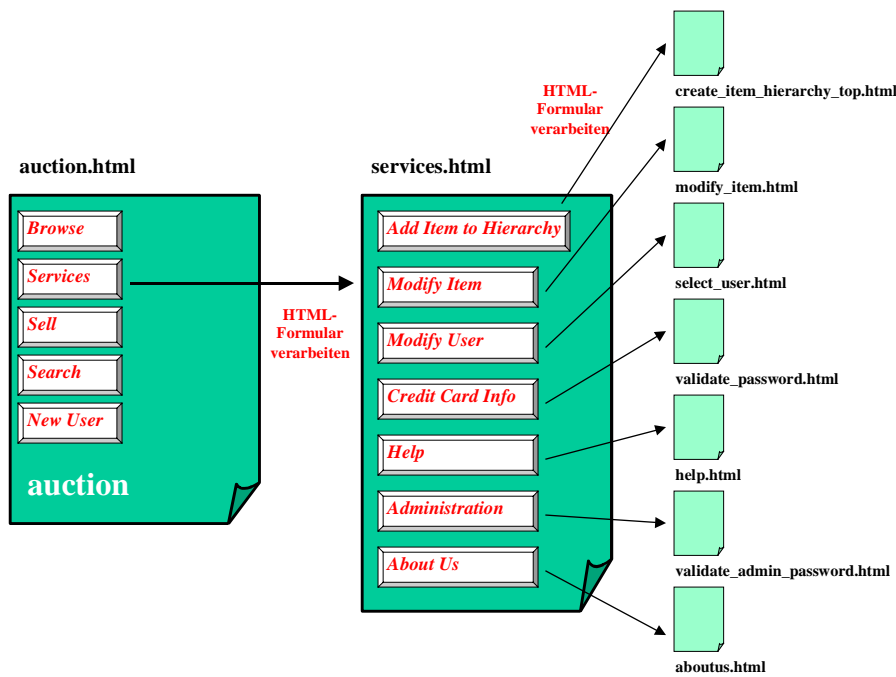
ein Gebot für das Objekt abzugeben, sich als neuer Benutzer registrieren zu lassen (sofern er noch nicht registriert ist), und auf die Homepage (`auction.html`) zurückzukehren.

- Eine Drucktaste anbieten, um durch die Kategorien zu blättern, in die die Datenbank eingeteilt ist.

Um alle zum Verkauf stehenden Objekte auf dem nächsten Template `display_items.html` anzuzeigen, benötigen Sie einen Baustein, der alle Objekte aus der Datenbank holt und sie anzeigt. In der Beispielanwendung übernimmt dies der Funktionsbaustein `AUCTION_DISPLAY_ITEMS`.

Jede Objekt-ID könnte dabei als Parameter zum Beschaffen von Einzelheiten über das Objekt dienen.

Services



Wenn der Benutzer *Services* wählt, könnte das nächste Template `services.html` einen weiteren Satz Drucktasten anbieten:

- *Add Item to Hierarchy*
Diese Drucktaste zeigt ein Template, in dem Benutzer Objekte zu Objektkategorien hinzufügen können.
- *Modify Item*
Diese Drucktaste zeigt ein Template, in dem Benutzer Objekte ändern können, die Sie in der Datenbank angelegt haben. Die Anwendung muß dazu jeden Benutzer auf Gültigkeit prüfen und seine Berechtigung überprüfen.
- *Modify User*

Flow-Datei-Anwendung entwerfen

Diese Drucktaste zeigt ein Template, in dem Benutzer Ihre eigenen persönlichen Angaben einschließlich Zahlungsinformationen ändern können. Die Anwendung muß jeden Benutzer auf Gültigkeit prüfen und sicherstellen, daß alle Benutzer ausschließlich ihre eigenen Daten ändern können.

- *Credit Card Info*

Diese Drucktaste zeigt ein Template, in dem Benutzer Kreditkarteninformationen eingeben können. Die Anwendung muß jeden Benutzer auf Gültigkeit prüfen.

- *Help*

Diese Drucktaste zeigt ein Template, in dem Benutzer Hilfe bekommen.

- *Administration*

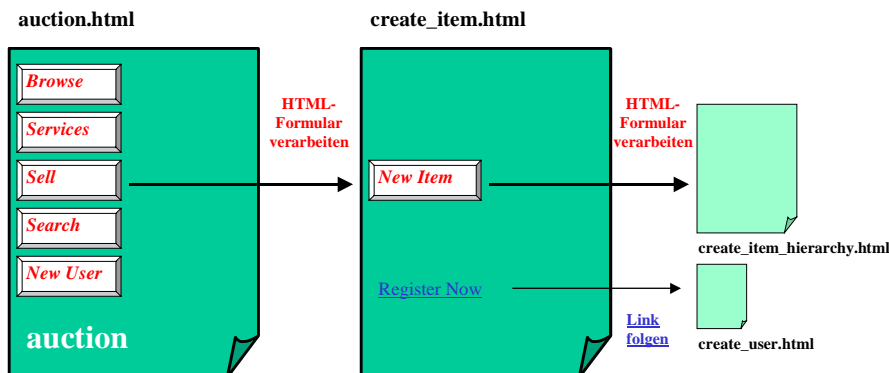
Diese Drucktaste zeigt ein Template mit einer Reihe von administrativen Funktionen. Um diese Funktionen einzusetzen, muß der Benutzer Administratorrechte haben.

- *About Us*

Diese Drucktaste zeigt ein Template, in dem allgemeine Informationen über die Firma angezeigt werden, die das Auktionsszenarium anbietet.

In diesem Fall ist keine Geschäftslogik erforderlich; Sie müssen also keine Funktionsbausteine aufrufen. Sie müssen nur auf das nächste Template `services.html` zeigen.

Sell



Wenn der Benutzer *Sell* wählt, könnte das nächste Template `create_item.html` Benutzern erlauben, ein neues Objekt zum Verkauf einzugeben und in der Datenbank abzulegen. Neue Benutzer müssen sich zuerst registrieren lassen, indem Sie den Link *Register Now* anklicken

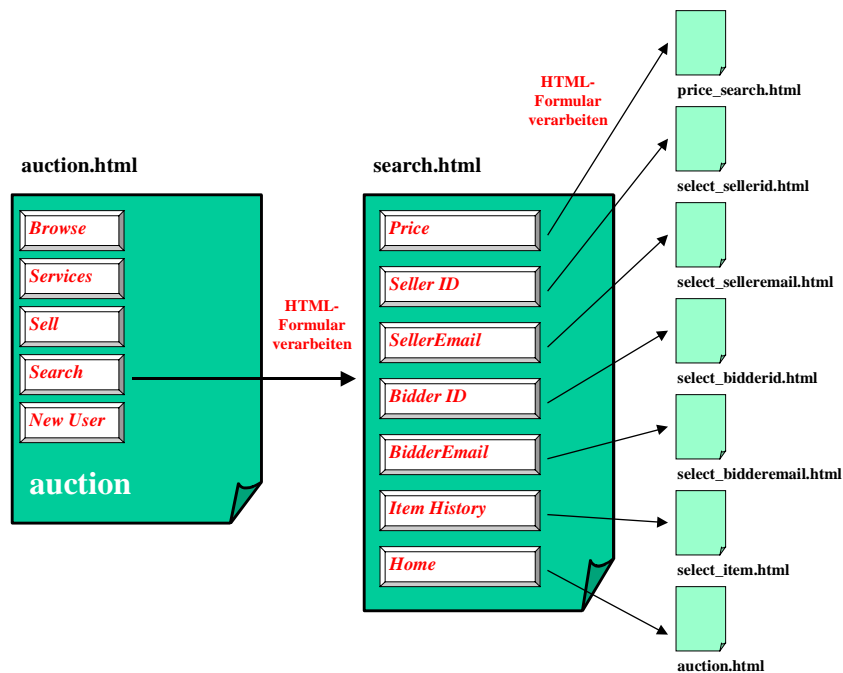
Flow-Datei-Anwendung entwerfen

Wenn Sie Benutzern erlauben, neue Objekte anzulegen, müssen Sie wissen, welche Benutzer bereits registriert sind und welche Objektkategorien es gibt. Dazu müssen Sie folgende Funktionsbausteine aufrufen:

- AUCTION_DISPLAY_BIDDERS
- AUCTION_DISPLAY_CATEGORIES

Zusätzlich müssen Sie eine Drucktaste vorsehen, mit der das Objekt in der Datenbank abgelegt werden kann, und ein Link, mit dem sich neue Benutzer registrieren lassen können.

Search



Wenn der Benutzer *Search* wählt, könnte das nächste Template `search.html` einen weiteren Satz Drucktasten anbieten:

- *Price*
Diese Drucktaste zeigt ein Template, in dem Benutzer innerhalb eines bestimmten Preisrahmens nach Objekten suchen können.
- *Seller ID*
Diese Drucktaste zeigt ein Template, in dem Benutzer über die Verkäufer-ID nach Verkäufern suchen können.
- *Seller Email*
Diese Drucktaste zeigt ein Template, in dem Benutzer über die e-Mail-Adresse nach Verkäufern suchen können.
- *Bidder ID*

Flow-Datei-Anwendung entwerfen

Diese Drucktaste zeigt ein Template, in dem Benutzer über die Bieter-ID nach Bietern suchen können.

- Bidder Email

Diese Drucktaste zeigt ein Template, in dem Benutzer über die e-Mail-Adresse nach Bietern suchen können.

- Item History

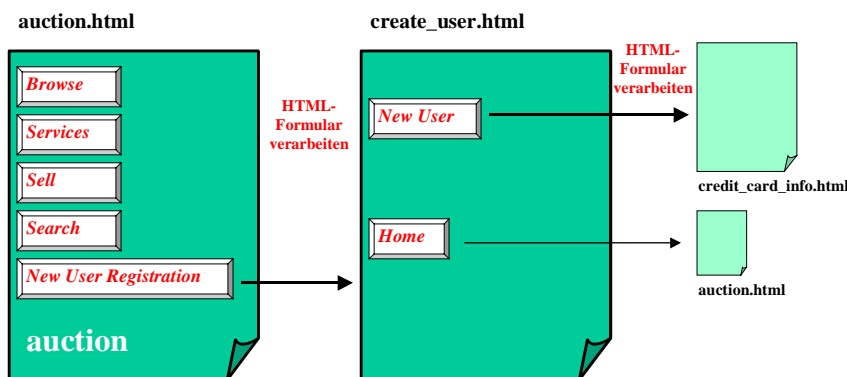
Diese Drucktaste zeigt ein Template, das die Historie jedes Objekts in der Datenbank anzeigt, einschließlich Detailinformationen wie Objektname, Objektbeschreibung, Preis und den letztmöglichen Termin (Datum und Uhrzeit) für Gebote.

- Home

Diese Drucktaste bringt Benutzer zurück auf die Homepage des Auktionsszenariums.

In diesem Fall ist keine Geschäftslogik erforderlich; Sie müssen also keine Funktionsbausteine aufrufen. Sie müssen nur auf das nächste Template `search.html` zeigen.

New User Registration



Wenn der Benutzer *New User Registration* wählt, muß das nächste Template `create_user.html` es neuen Benutzern ermöglichen, sich registrieren zu lassen. Es sollte auch eine Drucktaste anbieten, die Benutzer auf die Homepage des Auktionsszenariums bringt.

In diesem Fall ist keine Geschäftslogik erforderlich; Sie müssen also keine Funktionsbausteine aufrufen. Sie müssen nur auf das nächste Template `create_user.html` zeigen.

Flow-Datei-Anwendung entwerfen

Flow-Logik

Um den Dialogablauf der Anwendung abhängig von der an einer bestimmten Stelle vom Benutzer gewählten Aktion festzulegen, haben einige Templates Flow-Dateien, in denen Sie die Flow-Logik definieren.

In der Auktionsbeispielanwendung hat die Flow-Datei für das Ausgangs-Template `auction.html` den Namen `auction.flow`.

Ergebnis

Nachdem Sie die Anwendung entworfen haben, können Sie [die Geschäftslogik im R/3 definieren \[Seite 21\]](#).

Geschäftslogik im R/3 definieren

Verwendung

Um die Geschäftslogik in Ihre Flow-Datei-Anwendung zu implementieren, können Sie einen Satz remote-fähiger Funktionsbausteine im R/3-System anlegen.

Voraussetzungen

Sie haben Ihre [Anwendung entworfen \[Seite 14\]](#).

Vorgehensweise

1. In der ABAP Workbench legen Sie eine Entwicklungsklasse an (falls erforderlich).
2. Im Function Builder legen Sie eine Funktionsgruppe für Ihre Anwendung an.
3. Legen Sie einen Funktionsbaustein an.

Um alle Objekte in der Datenbank zu holen und anzuzeigen, verwendet die Beispielanwendung einen Funktionsbaustein namens AUCTION_DISPLAY_ITEMS.

Hier ist das Coding:

```
FUNCTION AUCTION_DISPLAY_ITEMS.  
*"-----  
*"Lokale Schnittstelle:  
*" EXPORTING  
*" VALUE(RETURNSTATUS) TYPE BAPIRETURN  
*" TABLES  
*" IAUCTIONITEMS STRUCTURE AUCTIONITEMS  
*"-----  
select * from AUCTIONITEMS  
into table IAUCTIONITEMS  
order by primary key.  
if sy-subrc <> 0.  
RETURNSTATUS-TYPE = 'E'.  
exit.  
endif.  
RETURNSTATUS-TYPE = 'S'."alles ok  
ENDFUNCTION.
```

Geschäftslogik im R/3 definieren

Ergebnis

Sie haben die Geschäftslogik für Ihre Anwendung definiert. Nun können Sie den Rest der Anwendung im SAP@Web Studio entwerfen. Beginnen Sie mit der [Einstellung Ihrer Umgebung \[Seite 23\]](#).

Danach können Sie die Präsentationsaspekte Ihrer Anwendung entwerfen. Legen Sie dazu HTML^{Business}-Templates an und definieren Sie den Dialogablauf in Flow-Dateien.



Weitere Informationen zum Anlegen von Funktionsbausteinen im Function Builder und zum Einsatz anderer Werkzeuge in der ABAP Workbench finden Sie unter [ABAP Workbench: Werkzeuge \[Extern\]](#).

Wenn dieser Link von Ihrem aktuellen Arbeitsplatz aus nicht funktioniert, sind Sie vielleicht gerade nicht im R/3-System. In diesem Fall melden Sie sich im neusten Release des R/3-Systems an und wählen Sie *Hilfe* → *SAP-Bibliothek* → *Basis* → *ABAP Workbench (BC-DWB)* → *BC – ABAP Workbench: Werkzeuge*.

Eine Liste aller in der Beispielanwendung verwendeten Funktionsbausteine finden Sie in [Anhang A: Funktionsbausteine in der Beispielanwendung \[Seite 41\]](#) am Ende dieses Tutorials.

SAP@Web Studio einrichten

Einsatzmöglichkeiten

Um die zur Ausführung Ihrer Anwendung erforderlichen Dateien des Internet Transaction Server (ITS-Dateien) anzulegen – Servicedatei, HTML^{Business}-Templates und Flow-Dateien –, können Sie das SAP@Web Studio verwenden.

Nachdem Sie alle diese Dateien angelegt haben, können Sie sie an eine ITS-Site publizieren und in die ITS-Source-Control im R/3-System hochladen. Dort können Sie sie, wie jedes andere Entwicklungsobjekt auch, einem Änderungsauftrag zuordnen.

Dieser Ablauf beschreibt, wie Sie die Umgebung im SAP@Web Studio einrichten.

Voraussetzungen

Das SAP@Web Studio muß installiert sein.

Ablauf

So richten Sie die Umgebung des SAP@Web Studio ein:

1. [Starten Sie das SAP@Web Studio \[Seite 24\]](#).
2. [Legen Sie ein ITS-Projekt an \[Seite 25\]](#).
3. [Definieren Sie eine ITS-Site \[Seite 26\]](#).

Ergebnis

Nachdem Sie die Umgebung eingerichtet haben, können Sie damit beginnen, die anderen ITS-Dateien anzulegen, die Sie zur Ausführung Ihrer Anwendung benötigen. Beginnen Sie mit dem [Anlegen eines ITS-Service \[Seite 28\]](#).

SAP@Web Studio starten

SAP@Web Studio starten

Um die Dateien des Internet Transaction Server (ITS-Dateien) anzulegen, die Sie zur Ausführung Ihrer Anwendung benötigen, müssen Sie das SAP@Web Studio starten und Ihre Umgebung einrichten.

Voraussetzungen

Das SAP@Web Studio muß installiert sein.

Vorgehensweise

Auf dem Windows NT-Desktop wählen Sie *Start* → *Programms* → *SAP@Web Studio* → *Studio* <release number>.

Ergebnis

Das SAP@Web Studio wird auf Ihrem Desktop geöffnet. Nun können Sie ein [ITS-Projekt anlegen](#) [Seite 25].

ITS-Projekt anlegen

Solange Sie eine Internet-Anwendung im SAP@Web Studio entwickeln, können Sie alle zusammengehörenden Dateien für die Dauer ihrer Bearbeitung in einem ITS-Projekt ablegen.

Diese Vorgehensweise beschreibt, wie Sie ein ITS-Projekt anlegen.

Vorgehensweise

1. Starten Sie das SAP@Web Studio wie unter [SAP@Web Studio starten \[Seite 24\]](#) beschrieben.
2. Wählen Sie *File* → *New* und dann das Register *Projects*.
3. Geben Sie im Feld *Project Name* einen Namen für das Projekt ein und wählen Sie *OK*.

Das SAP@Web Studio legt Ihr Projekt an und öffnet es automatisch in der Sicht *File View*.

Wenn Sie das SAP@Web Studio schließen und später mit der Arbeit an Ihrer Anwendung weitermachen wollen, können Sie das Projekt auf zwei Arten öffnen:

- Wählen Sie *File* → *Open Project* und suchen Sie im Dateisystem nach dem Projekt.
- Wählen Sie *File* → *Recent Projects* und öffnen Sie das Projekt von dieser Liste aus.

Wenn Sie ein Projekt öffnen, wird es zur Standardablage für alle Dateien, die Sie danach anlegen.

Ergebnis

Sie haben ein ITS-Projekt angelegt. Nun [definieren Sie eine ITS-Site \[Seite 26\]](#).

ITS-Site definieren

ITS-Site definieren

Verwendung

Wenn Sie eine Internet-Anwendung im SAP@Web Studio entwickeln, müssen Sie eine ITS-Site definieren.

Eine ITS-Site ist die Stelle, an die Sie alle im SAP@Web Studio angelegten Dateien publizieren (übergeben), um eine Anwendung zu implementieren.

- Die Stelle ist ein R/3-System.
- Die Dateien sind ITS-Dateien.
 - Dynamische Dateien wie Servicebeschreibungen, HTML^{Business}-Templates, Flow-Dateien und Sprachressourcen werden in den entsprechenden ITS-Serververzeichnissen abgelegt.
 - Statische MIME-Dateien (Multipurpose Internet Mail Extension) mit Grafiken, Bild-, Ton- und Video-Elementen werden im Web-Serververzeichnis abgelegt.

Während der Entwicklungsphase verwendet das SAP@Web Studio die ITS-Site, um auf alle Informationen zuzugreifen, die es zum Erzeugen eines Service benötigt.

Voraussetzungen

Sie haben [ein ITS-Projekt \[Seite 25\]](#) zum Speichern Ihrer Dateien angelegt.

Vorgehensweise

So definieren Sie eine ITS-Site:

1. Starten Sie das SAP@Web Studio wie unter [SAP@Web Studio starten \[Seite 24\]](#) beschrieben.
2. Wählen Sie *Project* → *Site Definition*.
3. Wählen Sie *New*.
4. Geben Sie einen Site-Namen ein und wählen Sie *Next*.
5. Geben Sie einen Web-Server-Hostnamen ein und wählen Sie *Next*.
6. Geben Sie den ITS-Hostnamen ein und wählen Sie *Next*.
7. Geben Sie den Namen einer virtuellen ITS-Instanz ein und wählen Sie *Next*.
8. Geben Sie bei Bedarf zusätzliche Web-Server-Informationen ein.
9. Wählen Sie *Finish* zur Bestätigung.
10. Wählen Sie *OK* zum Beenden.

Ergebnis

Das SAP@Web Studio zeigt die ITS-Site in der Listbox auf der linken Seite der Anwendungsfunktionsleiste an. Sie können nun die zur Ausführung Ihrer Anwendung

erforderlichen ITS-Dateien anlegen. Beginnen Sie mit dem [Anlegen einer ITS-Service-datei \[Seite 28\]](#).

ITS-Servicedatei anlegen

ITS-Servicedatei anlegen

Verwendung

Um Ihre Anwendung auszuführen, müssen Sie einen Service des Internet Transaction Server (ITS-Service) anlegen. Vielleicht möchten Sie auch eine Theme angeben.

- Ein ITS-Service ist die Menge externer Komponenten, die der ITS zur Ausführung einer Anwendung benötigt.
Die Servicedatei einer Flow-Datei-Anwendung muß eine Servicebeschreibung, HTML^{Business}-Templates und Flow-Dateien enthalten. Sie kann zusätzlich Sprachressourcen und MIME-Dateien (Multipurpose Internet Mail Extension) enthalten. Jeder ITS-Service wird durch die ITS-Servicebeschreibung definiert, die die wichtigsten Parameter für die Ausführung der Anwendung angibt.
- Eine ITS-Theme ist eine Instanz eines ITS-Service mit einem eigenen Satz HTML^{Business}-Templates und Sprachressourcen. Eine ITS-Theme gibt einem ITS-Service ein bestimmtes Erscheinungsbild, wobei die Funktionalität dieselbe bleibt. Wenn Sie nicht selbst eine andere Theme in der Servicedatei angeben, verwendet die Anwendung den Standardwert 99, der in der globalen Servicedatei `global.srvc` definiert ist.

Zur Laufzeit vermischt der ITS die Werte, die Sie in der Servicedatei angeben, mit den Werten aus der globalen Servicedatei, die Einstellungen enthält, die für alle Services gelten. Die Werte in der service-spezifischen Servicedatei überschreiben die in der globalen Servicedatei. Wenn in beiden Dateien wichtige Anmeldedaten fehlen (wie Mandant, Benutzername, Paßwort oder Anmeldesprache), zeigt der ITS das Anmeldebild an, damit der Benutzer die fehlenden Werte eintragen kann.

Voraussetzungen

Sie haben ein [ITS-Projekt angelegt \[Seite 25\]](#) und eine [ITS-Site definiert \[Seite 26\]](#).

Vorgehensweise

So legen Sie einen ITS-Service im SAP@Web Studio an:

1. Wenn das SAP@Web Studio noch nicht läuft, starten Sie es wie unter [SAP@Web Studio starten \[Seite 24\]](#) beschrieben.
2. Wählen Sie *File* → *New* und dann das Register *Files*.
3. Wählen Sie *Service File*.
4. Im Feld *Service Name* geben Sie einen Servicenamen für Ihre Anwendung ein und wählen *OK*.

Das SAP@Web Studio legt den Service in der Sicht *File View* des Projektarbeitsbereichs an und öffnet den Grid-Control-Editor, in dem Sie die erforderlichen Parameter angeben können.

5. In der Spalte *Name* stellen Sie den Cursor auf die erste leere Zeile und klicken zweimal.
Sie sehen den Standardnamen `~param1`, den Sie überschreiben können. Wenn Sie einen Parameternamen eingegeben haben, können Sie mit der *Tab*-Taste in die Spalte *Value* springen und einen entsprechenden Wert eingeben.

6. Geben Sie folgende Parameter und Parameterwerte ein:

Parameter	Wert	Beschreibung
<code>~xgateway</code>	<code>sapxginet</code>	XGateway ist eine Plug-In-Schnittstelle, die die diversen ITS-Implementierungsmodelle unterstützt. Dieser Parameter muß auf den Wert <code>sapxginet</code> gesetzt werden.
<code>~initialTemplate</code>	<code><template name></code>	Dies ist das erste Template, das beim Ausführen Ihrer Anwendung angezeigt wird. Der Name des Ausgangs-Template in der Beispielanwendung ist <code>auktion</code> .

Um eine Anwendung mit Flow-Dateien auszuführen, **muß** Ihre Servicedatei die oben genannten Parameter enthalten. Wenn Sie sie nicht angeben, wird die Anwendung nicht laufen.

7. Geben Sie nach Bedarf andere Parameter ein.

Sie können auch service-spezifische System- und Anmeldeinformationen angeben; diese Parameter sind jedoch optional.

8. Sichern Sie.

Ergebnis

Das SAP@Web Studio legt eine Servicedatei an, die alle vom ITS zur Ausführung Ihrer Anwendung benötigten Daten enthält.

So sieht die Servicedatei aus, wenn Sie nur die Minimaldaten enthält, die zur Ausführung der Auktionsbeispielanwendung erforderlich sind:

Parameter	Wert
<code>~xgateway</code>	<code>sapxginet</code>
<code>~initialTemplate</code>	<code>Auktion</code>

Nun [definieren Sie die Präsentation \[Seite 30\]](#).

Präsentation definieren

Präsentation definieren

Verwendung

Um eine Benutzungsoberfläche für Ihre Flow-Datei-Anwendung zu entwerfen, müssen Sie HTML^{Business}-Templates für den Service des Internet Transaction Server (ITS-Service) anlegen, den Sie gerade angelegt haben.

HTML^{Business}-Templates bestimmen das Erscheinungsbild Ihrer Anwendung. Es gibt ein Template für jedes Anwendungsdynpro.

HTML^{Business}-Templates enthalten Standard-HTML-Coding und HTML^{Business}-Anweisungen. Wenn ein Benutzer eine Anwendung startet, bei der die Bausteine, die die Geschäftslogik definieren, im R/3-System liegen, mischt der ITS R/3-Daten, R/3-Texte, R/3-Fehlermeldungen, Nicht-R/3-Texte und Grafiken sowie dynamisch generierte URLs in ein Template ein, indem er die HTML^{Business}-Anweisungen interpretiert. Das fertige Template wird im Web-Browser des Benutzers angezeigt.

Voraussetzungen

Sie haben bereits eine [ITS-Servicedatei angelegt \[Seite 28\]](#).

Vorgehensweise

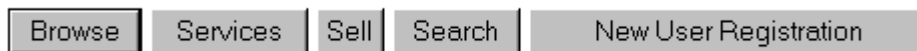
So legen Sie HTML^{Business}-Templates für Ihre Anwendung an:

1. Wenn das SAP@Web Studio noch nicht läuft, starten Sie es wie unter [SAP@Web Studio starten \[Seite 24\]](#) beschrieben.
2. Wählen Sie *File* → *New* und dann das Register *Files*.
3. Wählen Sie *HTML Business Template*.
4. Im Feld *Name* geben Sie einen Template-Namen ein und wählen *OK*.
5. Geben Sie das HTML^{Business}-Coding ein, das das benötigte Template-Layout definiert.
6. Sichern Sie.

Ergebnis

Sie haben das Ausgangs-Template für Ihre Anwendung angelegt. Nun [definieren Sie die Flow-Logik \[Seite 32\]](#).

Das Ausgangs-Template in der Beispielanwendung zeigt fünf Drucktasten:



Hier ist das HTML^{Business}-Coding für das Ausgangs-Template:

```
<html>
```

```
<form action="`wgateURL()`" method="post" target="_blank" >
```

```
<table>
  <tr>
    <td align=left>
      <input type="submit" name="~event" value="Browse">
    </td>
    <td align=left>
      <input type="submit" name="~event" value="Services">
    </td>
    <td align=left>
      <input type="submit" name="~event" value="Sell">
    </td>
    <td align=left>
      <input type="submit" name="~event" value="Search">
    </td>
    <td align=left>
      <input type="submit" name="~event" value="New User
Registration">
    </td>
  </tr>
</table>

</form>

</html>
```



Obwohl es sich anbietet, den Editor im SAP@Web Studio zu verwenden, können Sie HTML^{Business}-Templates mit jedem geeigneten HTML-Editor anlegen.

Flow-Logik definieren

Flow-Logik definieren

Verwendung

Jedes HTML^{Business}-Template in Ihrer Flow-Datei-Anwendung, das eine Dialogablaufdefinition benötigt, um auf Benutzeraktionen zu reagieren, muß eine zugeordnete Flow-Datei haben, die die Flow-Logik enthält.

Die Flow-Logik definiert logische Übergänge zwischen Anwendungszuständen, abhängig von der vom Benutzer gewählten Aktion.

Wenn ein HTML^{Business}-Template den Namen `<template name>.html` hat, heißt die zugeordnete Flow-Datei (wenn es eine gibt) `<template name>.flow`. Diese Flow-Datei enthält einen Satz Ereignisse und Zustände. Ereignisse definieren die Eintrittspunkte von unterschiedlichen logischen Zuständen, oder wie ein anderes Template direkt geladen wird.

Sie definieren die Flow-Logik mit Hilfe eines Subset von XML-Elementen (Extensible Markup Language).

Voraussetzungen

Sie haben die [Präsentation definiert \[Seite 30\]](#), indem Sie ein HTML^{Business}-Template angelegt haben.

Vorgehensweise

1. Wenn das HTML^{Business}-Template, für das Sie eine Flow-Datei erzeugen wollen, noch nicht im Hauptarbeitsbereich des HTML^{Business}-Editors geöffnet ist, öffnen Sie es durch Doppelklick.

Das SAP@Web Studio öffnet das Template im Hauptarbeitsbereich.

2. Wählen Sie *Add Flow File* oder *Edit* → *Flow File*.

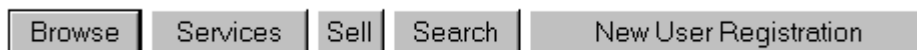
Das SAP@Web Studio legt eine Flow-Datei an und öffnet sie mit den Anfangs- und Endebefehlen `<flow>` und `</flow>`.

Sie können zwischen dem im HTML^{Business}-Template definierten HTML^{Business}-Coding und der in der Flow-Datei definierten Flow-Logik hin- und herschalten. Verwenden Sie dazu die Tasten *HTML Source* und *Flow Source* in der Statusleiste des HTML^{Business}-Editors.

3. Geben Sie zwischen den Befehlen `<flow>` und `</flow>` die Flow-Logik ein.

Für das Ausgangs-Template `action.html` der Auktionsbeispielanwendung benötigen Sie eine Flow-Datei, die je nach vom Benutzer gewählter Aktion die entsprechenden Bausteine ausführt.

`action.html` enthält fünf Ereignisse, die durch die Drucktasten dargestellt werden: *Browse* ("Blättern"), *Services*, *Sell* ("Verkaufen"), *Search* ("Suchen") und *New User Registration* ("Neuen Benutzer registrieren").



Einige dieser Drucktasten führen direkt auf das nächste Template. Andere enthalten Verarbeitungsblöcke.

Flow-Logik definieren

- Wenn der Benutzer *Browse* wählt, instruiert die Flow-Logik den ITS, den Baustein `auction_display_items` auszuführen, der alle Objekte von der Datenbank holt.
 - Wenn es Objekte gibt, erhält die Anwendung das nächste Template `display_items.html` und zeigt die Objekte an.
 - Wenn es keine Objekte gibt, holt das ITS das Standard-Template `empty.html` und zeigt es an.
- Wenn der Benutzer *Services* wählt, müssen keine Daten angezeigt werden. Daher instruiert die Flow-Logik den ITS, das nächste Template `services.html` anzuzeigen.
- Wenn der Benutzer *Sell* wählt, instruiert die Flow-Logik den ITS, folgende Bausteine auszuführen:
 - `auction_display_categories`: holt alle gegenwärtig in der Datenbank gespeicherten Kategorien und zeigt sie an.
 - `auction_display_bidders`: holt alle gegenwärtig in der Datenbank gespeicherten Bieter.

Die Flow-Logik instruiert den ITS auch, das Template `create_item.html` anzuzeigen.
- Wenn der Benutzer *Search* wählt, müssen keine Daten angezeigt werden. Daher instruiert die Flow-Logik den ITS, das nächste Template `search.html` anzuzeigen.
- Wenn der Benutzer *New User Registration* wählt, müssen keine Daten angezeigt werden. Daher instruiert die Flow-Logik den ITS, das nächste Template `create_user.html` anzuzeigen.

Hier ist das Coding für die Flow-Datei `auction.flow`, die das Ausgangs-Template `auction.html` steuert:

```

<flow>

<event name = "Browse" next_state="Browse">
</event>

<state name = "Browse">
  <module name= "auction_display_items" type="RFC">
    <default next_template = "empty"></default>
    <result next_template = "display_items">
      <expr>IAUCTIONITEMS-ITEMID.dim > 0 </expr>
    </result>
  </module>
</state>

<event name = "Services" next_template="Services">
</event>

<event name = "Sell" next_state="create_item">
</event>

<state name="create_item">
  <module name= "auction_display_categories" type="RFC">
    </module>

```

Flow-Logik definieren

```
<module name= "auction_display_bidders" type="RFC">
  <default next_template = "CREATE_ITEM"></default>
</module>
</state>

<event name = "Search" next_template="SEARCH">
</event>

<event name = "New User Registration"
next_template="create_user">
</event>

</flow>
```

4. Sichern Sie.

Ergebnis

Sie haben eine Flow-Datei für Ihr HTML^{Business}-Template angelegt. Nun [legen Sie die übrigen Templates \[Seite 30\]](#) und Flow-Dateien an.

Nachdem Sie alle HTML^{Business}-Templates und Flow-Dateien angelegt haben, gehen Sie so vor:

1. Bei Bedarf [legen Sie andere ITS-Dateien an \[Seite 35\]](#), die Sie für Ihre Anwendung brauchen.

Diese Dateien können auch Sprachressourcen und MIME-Dateien sein.

2. [Publizieren Sie die Dateien \[Seite 36\]](#) an eine ITS-Site.

Eine Liste aller XML-Elemente, die Sie zur Definition der Flow-Logik in Flow-Dateien verwenden können, finden Sie unter [Flow-Logik-Syntax \[Extern\]](#).

Andere ITS-Dateien anlegen

Wenn Sie Ihre Anwendung entwickeln, möchten Sie vielleicht auch andere ITS-Dateien einbeziehen, wie etwa:

- Sprachressourcen
- MIME-Dateien (Multipurpose Internet Mail Extension)

Obwohl diese Dateien die Performance und Erscheinung Ihrer Anwendung verbessern, sind sie nicht vorgeschrieben.

Sprachressourcen

Um den ITS-Service, der Ihre Anwendung ausführt, sprachenunabhängig zu halten, können Sie Sprachressourcen anlegen.

Sprachressourcendateien enthalten alle Texte, die erforderlich sind, um einen ITS-Service in einer bestimmten Sprache auszuführen. Für jeden Text geben Sie einen Platzhalter und den zugehörigen Text in der Sprachressourcendatei an. In den HTML^{Business}-Templates verwenden Sie dann die Platzhalter anstelle der eigentlichen Texte. Dies bedeutet, daß Sie für alle Sprachen zusammen nur einen einzigen Satz Templates anlegen müssen, anstatt einen Satz pro Sprache. Zur Laufzeit sucht der ITS die Platzhalter in jedem Template und ersetzt sie durch den richtigen Sprachentext aus der Sprachressourcendatei.



Nicht alle ITS-Services verwenden Sprachressourcen. Einige HTML^{Business}-Templates enthalten hartkodierte Sprachentexte und sind daher sprachenspezifisch. Um Ihren ITS-Services so flexibel wie möglich zu halten, sollten Sie versuchen, bei jeder passenden Gelegenheit mit Sprachressourcen zu arbeiten.

Solange Sie nicht mit sprachenspezifischen HTML^{Business}-Templates arbeiten, sollten Sie Sprachressourcendateien für die Sprachen erstellen, in denen Sie Ihre Anwendung entwickeln.

MIME-Dateien

MIME-Dateien enthalten die Grafiken, Bild- und Video-Elemente, die zur Verbesserung oder Personalisierung Ihrer Anwendung dienen.

Vorgehensweise

Weitere Informationen über das Anlegen von Sprachressourcen und MIME-Dateien finden Sie in der [SAP@Web Studio \[Extern\]](#)-Dokumentation.

Ergebnis

Nachdem Sie alle ITS-Dateien angelegt haben, die Sie für Ihre Anwendung brauchen, [publizieren \[Seite 36\]](#) Sie sie an eine ITS-Site.

ITS-Service publizieren

ITS-Service publizieren

Nachdem Sie alle Dateien des Internet Transaction Server (ITS-Dateien) angelegt haben, die Ihre Anwendung bilden, können Sie den Service an die zuvor definierte [ITS-Site \[Seite 26\]](#) publizieren.

Wenn Sie der unten beschriebenen Vorgehensweise folgen, können Sie die Anwendung sofort ausführen.

Vorgehensweise

So publizieren Sie die ITS-Dateien, die Ihre Anwendung bilden, an eine ITS-Site und testen Sie sofort:

1. Wählen Sie *Project* → *Set Active Service* und wählen Sie den entsprechenden Service aus.

So wird der ausgewählte Service zum gerade aktiven Service. Dies aktiviert die Tasten *Publish*, *Build* und *Go* in der Publizierleiste direkt über dem Arbeitsbereich.

2. Wählen Sie *Options* → *Studio Properties* und dann das Register *Web Browser*.
3. Markieren Sie *Open Browser window in Studio* (wenn nicht bereits markiert) und wählen Sie *OK*.

Dadurch können Sie die Anwendung direkt aus dem SAP@Web Studio ausführen.

4. Wählen Sie *Go*.

Wenn Sie dies wählen, führt das SAP@Web Studio folgende Schritte durch:

- Publizieren des Service an die gerade aktive ITS-Site.
- Öffnen eines Web-Browser.
- Starten der Anwendung.

Ergebnis

Wenn die Anwendung startet, führt das SAP@Web Studio folgendes aus:

- Anzeige des Ausgangs-Template Ihrer Anwendung im Fenster des Web-Browser.
Das Ausgangs-Template ist das Template, das Sie im Parameter `-initialTemplate` in der Servicedatei Ihrer Anwendung angegeben haben.
Das Ausgangs-Template der Beispielanwendung ist `auction.html`.
- Anzeige eines kurzen Protokolls im Nachrichtenbereich direkt unterhalb des Arbeitsbereichs.

Wenn Sie die Anwendung nicht ausführen wollen, können Sie auch die Tasten *Build* und *Publish* in der Publizierleiste des SAP@Web Studio verwenden:

- Um den gesamten Service an die gerade aktive ITS-Site zu publizieren, wählen Sie *Build*.
- Um einzelne Dateien an angegebenen Sites zu publizieren, wählen Sie *Publish*.

Nun [laden Sie die Dateien in die ITS-Source-Control \[Seite 38\]](#) im R/3-System.

Weitere Informationen über das Publizieren und Testen von ITS-Dateien finden Sie unter [ITS-Service publizieren \[Extern\]](#) in der SAP@Web Studio-Dokumentation.

Dateien in die ITS-Source-Control laden

Dateien in die ITS-Source-Control laden

Nachdem Sie den Service publiziert und getestet haben, können Sie alle Dateien des Internet Transaction Server (ITS-Dateien) in die ITS-Source-Control im R/3-System laden. Dadurch schützen Sie die Dateien vor gleichzeitiger Änderung durch mehrere Benutzer, da alle Benutzer zwar Lesezugriff haben, aber nur ein einziger Benutzer die Dateien jeweils herunterladen und ändern darf.

Vorgehensweise

So laden Sie die Dateien in die ITS-Source-Control:

1. Melden Sie sich am R/3-System an, in das Sie die Dateien hochladen wollen.
 - a. Wählen Sie *Tools* → *ITS source control* → *Connect to R/3*.
 - b. Markieren Sie ein R/3-System und wählen Sie *OK*.
 - c. Bei Bedarf geben Sie Anmeldeinformationen ein (Mandant, Benutzer, Paßwort, Anmeldsprache).
 - d. Wählen Sie *OK*.
2. Laden Sie die Dateien hoch.
 - a. Im Projektarbeitsbereich des SAP@Web Studio markieren Sie die Datei(en), die Sie hochladen wollen.
 - b. Klicken Sie mit der rechten Maustaste und wählen Sie *Add to ITS source control*.
 - c. Überprüfen Sie die markierten Dateien und wählen Sie *OK*.
3. Ordnen Sie die Dateien einem Änderungsauftrag zu.

Wenn Sie dies nicht tun, können Sie die Dateien für spätere Änderungen nicht mehr herunterladen.

 - a. Wählen Sie *Werkzeuge* → *Web-Entwicklung* → *Verwaltung Web-Objekte*.
 - b. Im Feld *Servicename* geben Sie den Servicenamen für die Dateien an, die Sie gerade hochgeladen haben.
 - c. Wählen Sie *Ausführen*.
 - d. Markieren Sie den Service und wählen Sie *Transport*.
 - e. Geben Sie eine gültige Änderungsauftragsnummer ein oder legen Sie einen neuen Auftrag im Transport Organizer an.

Ergebnis

Der Service wird einem Änderungsauftrag im R/3-System zugeordnet. Sie können die Dateien nun in der ITS-Source-Control verwalten. Sie können die Dateien auch herunterladen, ändern und wieder hochladen.

Bei späteren Hochladeaktionen erinnert sich das R/3-System an den entsprechenden Änderungsauftrag und ersetzt alte Dateiversionen durch neue.

Dateien in die ITS-Source-Control laden

Wenn Sie einen Änderungsauftrag zum Transport freigeben, müssen Sie einen neuen Änderungsauftrag anlegen. Wenn Sie dies nicht tun, können Sie die Dateien nicht mehr herunterladen.

Weitere Informationen über die Versionsverwaltung von ITS-Dateien in der ITS-Source-Control, einschließlich weiterer Einzelheiten zu den Hoch- und Herunterladevorgängen, finden Sie unter [ITS-Source-Control \[Extern\]](#) in der SAP@Web Studio-Dokumentation.

Flow-Datei-Anwendungen debuggen

Flow-Datei-Anwendungen debuggen

Verwendung

Um Internet-Anwendungen mit Flow-Dateien zu debuggen, müssen Sie den Debugging-Modus für die entsprechende ITS-Instanz aktivieren, den ITS-Debugger einrichten, RFC-Debugging in der Servicedatei aktivieren und Haltepunkte im ABAP-Coding setzen.

Denken Sie daran, daß die RFC-Verbindung zustandslos ist. Daher schließt der Debugging-Modus sobald Sie das entsprechende ABAP-Programm zuende ausgeführt haben.

Vorgehensweise

Um Ihre Anwendung auf Fehler zu untersuchen, führen Sie folgende Schritte aus:

1. Aktivieren Sie den Debugging-Modus für die entsprechende ITS-Instanz (wie unter [Debugging ein- und ausschalten \[Extern\]](#) beschrieben).
 - Sie (oder Ihr Systemadministrator) kann dies in der ITS-Verwaltung tun.
2. Richten Sie den ITS-Debugger ein. Melden Sie sich dazu an einer R/3-Instanz an, die Ihrem ITS-Anwendungsserver entspricht:
 - a. In der SAP-Anmeldebox wählen Sie *Neu*
 - b. Geben Sie in folgenden Feldern Werte ein:
 - *Beschreibung*
Verwenden Sie einen sprechenden Namen.
 - *Anwendungsserver*
Dies sollte derselbe wie Ihr ITS-Server sein.
 - *Systemnummer*
Der Standard ist 0.
 - c. Wählen Sie *OK*.
 - In der SAP-Anmeldebox wird ein neuer Eintrag angelegt.
3. In der Servicedatei Ihrer Anwendung setzen Sie den Parameter `~rfcDebugging` auf 1.
4. Im ABAP-Coding setzen Sie Haltepunkte, wo es erforderlich scheint.
5. Führen Sie die Anwendung aus wie unter [Tutorial - Flow-Datei-Anwendungen: Beispielanwendung \[Seite 10\]](#) beschrieben.

Ergebnis

Die Anwendung wird bei jedem von Ihnen gesetzten Haltepunkt unterbrochen.

Anhang A: Funktionsbausteine in der Beispielanwendung

Die zur Ausführung der Beispielanwendung benötigte Geschäftslogik wird in einem Satz remote-fähiger Funktionsbausteine in der Funktionsgruppe AUCTION_FUNCTIONS definiert.

Die folgende Tabelle zeigt einer Liste der Funktionsbausteine in Funktionsgruppe AUCTION_FUNCTIONS mit einer kurzen Beschreibung:

Funktionsbaustein	Beschreibung
AUCTION_BROWSE_ITEMS_BY_PRICE	Holt Objekte auf Preisbasis.
AUCTION_CREATE_BID	Legt Gebot für Objekt an.
AUCTION_CREATE_CATALOG	Legt Katalog an.
AUCTION_CREATE_CATALOG_HIER	Legt Kataloghierarchie an.
AUCTION_CREATE_CREDIT_CARD	Legt Kreditkarteninformation an.
AUCTION_CREATE_ITEM	Legt Objekt in Katalog(en) an.
AUCTION_CREATE_ITEMS_HIER	Legt Objekt unter Katalog(en) an.
AUCTION_CREATE_USER	Registriert einen Benutzer.
AUCTION_DISPLAY_BIDDERS	Zeigt Liste der registrierten Bieter an.
AUCTION_DISPLAY_CATEGORIES	Zeigt Kategorien an.
AUCTION_DISPLAY_CATEGORY_OBJS	Zeigt Unterkategorien und Objekte in einer Kategorie an.
AUCTION_DISPLAY_ITEM	Zeigt Objektdetails, höchstes Gebot und Verkäuferinformation an.
AUCTION_DISPLAY_ITEMS	Zeigt Objekte einer Kategorie an.
AUCTION_DISPLAY_ITEM_HISTORY	Zeigt Gebotshistorie für ein Objekt an.
AUCTION_DISPLAY_USER	Zeigt Informationen über einen registrierten Benutzer an.
AUCTION_GET_PAYMENT_INFO	Holt Kreditkarteninformationen.
AUCTION_MODIFY_ITEM	Ändert bestehendes Objekt.
AUCTION_MODIFY_PAYMENT_INFO	Ändert Kreditkarteninformationen eines Benutzers.
AUCTION_MODIFY_USER	Ändert Benutzerinformationen.
AUCTION_MODIFY_USER_ADMIN_USE	Administratorfunktion zum Ändern von Benutzerinformationen.
AUCTION_PURGE_ITEMS	Löscht verkaufte Objekte.
AUCTION_SEARCH_ITEMS_BY_BEMAIL	Sucht Objekte mit Gebot nach e-Mail-Adresse des Bieters.

Anhang A: Funktionsbausteine in der Beispielanwendung

AUCTION_SEARCH_ITEMS_BY_BIDDER	Sucht Objekte mit Gebot nach Bieter-ID.
AUCTION_SEARCH_ITEMS_BY_SELLER	Sucht zu verkaufende Objekte nach Verkäufer-ID.
AUCTION_SEARCH_ITEMS_BY_SEMAIL	Sucht zu verkaufende Objekte nach e-Mail-Adresse des Verkäufers.
AUCTION_VALIDATE_ADMIN_PSWD	Bestätigt Administratorpaßwort.
AUCTION_VALIDATE_USER_PSWD	Bestätigt registrierten Benutzer.

Anhang B: ITS-Dateien in der Beispielanwendung

Zur Ausführung der Beispielanwendung sind folgende ITS-Dateien erforderlich:

- Eine Servicedatei, die die zur Steuerung der Anwendung benötigten Parameter enthält.
- Ein Satz HTML^{Business}-Templates, die die Präsentation definieren.
- Ein Satz Flow-Dateien, die den Dialogablauf definieren.
 Jedem HTML^{Business}-Template, das eine Dialogablaufdefinition erfordert, ist eine Flow-Datei zugeordnet.

Servicedatei

Die Servicedatei heißt `auctionnew.srvc`.

HTML^{Business}-Templates und Flow-Dateien

Die folgende Tabelle zeigt eine Liste der HTML^{Business}-Templates und zugeordneten Flow-Dateien, die in der Beispielanwendung verwendet werden. Nicht alle Templates haben Flow-Dateien.

HTML ^{Business} -Template	Flow-Datei
<code>aboutus.html</code>	
<code>admin_modify_users.html</code>	<code>admin_modify_users.flow</code>
<code>administration.html</code>	<code>administration.flow</code>
<code>auction.html</code>	<code>auction.flow</code>
<code>auth_failed.html</code>	
<code>authenticate.html</code>	<code>authenticate.flow</code>
<code>bid_successful.html</code>	<code>bid_successful.flow</code>
<code>browse_items_by_price.html</code>	
<code>create_bid.html</code>	<code>create_bid.flow</code>
<code>create_catalog.html</code>	<code>create_catalog.flow</code>
<code>create_category_hierarchy.html</code>	<code>create_category_hierarchy.flow</code>
<code>create_category_hierarchy_top.html</code>	<code>create_category_hierarchy_top.flow</code>
<code>create_item.html</code>	<code>create_item.flow</code>
<code>create_item_hierarchy.html</code>	<code>create_item_hierarchy.flow</code>
<code>create_item_hierarchy_top.html</code>	<code>create_item_hierarchy_top.flow</code>
<code>create_user.html</code>	<code>create_user.flow</code>
<code>credit_card_info.html</code>	<code>credit_card_info.flow</code>
<code>credit_card_info1.html</code>	<code>credit_card_info1.flow</code>
<code>display_catalog.html</code>	<code>display_catalog.flow</code>

Anhang B: ITS-Dateien in der Beispielanwendung

display_category_objects.html	display_category_objects.flow
display_credit_card_detail.html	display_credit_card_detail.flow
display_credit_card_info.html	display_credit_card_info.flow
display_item.html	display_item.flow
display_item_history.html	display_item_history.flow
display_items.html	display_items.flow
display_items_bidderemail.html	
display_items_bidderid.html	display_items_bidderid.flow
display_items_sellerid.html	display_items_sellerid.flow
display_users.html	display_users.flow
displayarea.html	
empty.html	
enter_catalog.html	enter_catalog.flow
failed.html	
header.html	header.flow
help.html	
invaliduser.html	
modify_item.html	modify_item.flow
modify_user.html	modify_user.flow
modify_user_template.html	modify_user_template.flow
no_bids_made.html	
price_search.html	price_search.flow
pricefailed.html	
Purgecompletedsuccessfully.html	
registration.html	registration.flow
search.html	search.flow
select.html	select.flow
select_bidderemail.html	select_bidderemail.flow
select_bidderid.html	select_bidderid.flow
select_item.html	select_item.flow
select_seller.html	select_seller.flow
select_selleremail.html	select_selleremail.flow
select_sellerid.html	select_sellerid.flow
select_user.html	select_user.flow

Anhang B: ITS-Dateien in der Beispielanwendung

<code>services.html</code>	<code>services.flow</code>
<code>start.html</code>	<code>start.flow</code>
<code>suggestions.html</code>	
<code>validate_admin_password.html</code>	<code>validate_admin_password.flow</code>
<code>validate_password.html</code>	<code>validate_password.flow</code>