

HP Unified Functional Testing

ソフトウェア・バージョン : 11.51 Service Pack

API テスト・チュートリアル

ドキュメント・リリース日 : 2013 年 1 月 (英語版)

ソフトウェア・リリース日 : 2013 年 1 月 (英語版)



ご注意

保証

HP 製品、またはサービスの保証は、当該製品、およびサービスに付随する明示的な保証文によってのみ規定されるものとします。ここでの記載で追加保証を意図するものは一切ありません。ここに含まれる技術的、編集上の誤り、または欠如について、HP はいかなる責任も負いません。

ここに記載する情報は、予告なしに変更されることがあります。

権利の制限

機密性のあるコンピューターソフトウェアです。これらを所有、使用、または複製するには、HP からの有効な使用許諾が必要です。商用コンピューターソフトウェア、コンピューターソフトウェアに関する文書類、および商用アイテムの技術データは、FAR12.211 および 12.212 の規定に従い、ベンダーの標準商用ライセンスに基づいて米国政府に使用許諾が付与されます。

著作権について

© Copyright 2010 - 2012 Hewlett-Packard Development Company, L.P.

商標について

Java は、Oracle Corporation およびその関連会社の登録商標です。

Microsoft® および Windows® は、Microsoft Corporation の米国における登録商標です。

Oracle® は、Oracle Corporation (カリフォルニア州 Redwood City) の米国登録商標です。

Unix® は、The Open Group の登録商標です。

SlickEdit® は、SlickEdit Inc. の登録商標です。

ドキュメントの更新情報

このマニュアルの表紙には、以下の識別情報が記載されています。

- ソフトウェアバージョンの番号は、ソフトウェアのバージョンを示します。
- ドキュメントリリース日は、ドキュメントが更新されるたびに変更されます。
- ソフトウェアリリース日は、このバージョンのソフトウェアのリリース期日を表します。

更新状況、およびご使用のドキュメントが最新版かどうかは、次のサイトで確認できます。

<http://support.openview.hp.com/selfsolve/manuals>

このサイトを利用するには、HP Passportへの登録とサインインが必要です。HP Passport IDの登録は、次のWebサイトから行なうことができます。

<http://h20229.www2.hp.com/passport-registration.html> (英語サイト)

または、HP Passport のサインインページの [**New users - please register**] をクリックします。

適切な製品サポートサービスをお申し込みいただいたお客様は、更新版または最新版をご入手いただけます。詳細は、HPの営業担当にお問い合わせください。

サポート

次のHPソフトウェアサポートのWebサイトを参照してください。

<http://support.openview.hp.com>

このサイトでは、HPのお客様窓口のほか、HPソフトウェアが提供する製品、サービス、およびサポートに関する詳細情報をご覧いただけます。

HPソフトウェアオンラインではセルフソルブ機能を提供しています。お客様のビジネスを管理するのに必要な対話型の技術サポートツールに、素早く効率的にアクセスできます。HPソフトウェアサポートのWebサイトでは、次のようなことができます。

- 関心のあるナレッジドキュメントの検索
- サポートケースの登録とエンハンスメント要求のトラッキング
- ソフトウェアパッチのダウンロード
- サポート契約の管理
- HP サポート窓口の検索
- 利用可能なサービスに関する情報の閲覧
- 他のソフトウェアカスタマーとの意見交換
- ソフトウェアトレーニングの検索と登録

一部のサポートを除き、サポートのご利用には、HP Passport ユーザーとしてご登録の上、サインインしていただく必要があります。また、多くのサポートのご利用には、サポート契約が必要です。HP Passport ID を登録するには、次のWebサイトにアクセスしてください。

<http://h20229.www2.hp.com/passport-registration.html> (英語サイト)

アクセスレベルの詳細については、次のWebサイトをご覧ください。

http://support.openview.hp.com/access_level.jsp

目次

チュートリアルへようこそ	7
チュートリアルのレッスン	7
レッスン 1 : HP Unified Functional Testing - API テストの概要	9
SOA とは	9
SOA テストを自動化する理由	10
用語について	11
このチュートリアルのサンプル・アプリケーション	12
アプリケーションを起動する方法	13
レッスン 2 : 単純なテストの作成	15
新しいテストを作成する方法	16
Unified Functional Testing の表示枠を使用する方法	17
テスト・ステップを作成する方法	17
テスト・ステップを接続する方法	22
複数のソースからのデータをマップする方法	24
ステップをデータ駆動する方法	26
レッスン 3 : Web サービスのテスト	31
Web サービスをインポートする方法	32
Web サービス・テストを作成する方法	34
テストにデータを組み入れる方法	40
複数のデータ・ソースとカスタム・コードを使用する方法	45
レッスン 4 : REST サービスのテスト	51
REST サービス・アクティビティを作成する方法	52
REST テストを実行する方法	58
REST メソッドにデータを割り当てる方法	60
出力を確認する方法	65
REST サービスの変更を解決する方法	67

チュートリアルへようこそ

非 GUI テスト用の『API テスト Unified Functional Testing チュートリアル』へようこそ。これは、印刷可能な自習用のガイドで、Web サービス、REST サービスといった GUI のないアプリケーション用のテストの作成手順を説明するのが目的です。

このチュートリアルが終了すると、独自のアプリケーションについて簡単なテストを設計し、実行し、監視する準備が整ったこととなります。このチュートリアルは、その情報が記載されている順序どおりに進めることをお勧めします。

チュートリアルのレッスン

このチュートリアルのレッスンは次のとおりです。

レッスン 1 : HP Unified Functional Testing - API テストの概要 : API テスト用 HP Unified Functional Testing チュートリアルについて説明します。

レッスン 2 : 単純なテストの作成 : アクティビティをキャンバスにドラッグ・アンド・ドロップしてテストを作成する基本的な方法を紹介します。また、外部データでテストをデータ駆動する方法についても説明します。

レッスン 3 : Web サービスのテスト : Web サービスのテストを作成する方法と、結果をチェックする方法を示します。

レッスン 4 : REST サービスのテスト : REST サービスをモデル化し、そのテストを作成する方法を示します。

はじめに

レッスン 1

HP Unified Functional Testing - API テストの概要

API テスト用の HP Unified Functional Testing には、ヘッドレス・システム（ユーザ・インタフェースのないシステム）の機能テストを作成し実行するための拡張可能なフレームワークが含まれています。このドキュメントは、HP Unified Functional Testing を初めて使用し、独自のテストを初めて作成する方法について説明します。また、製品の主要な機能と、その機能をテストに組み込む方法も紹介します。

テストは、アクティビティをキャンバスにドラッグして作成します。パラメータを設定し、値を割り当てて、テストを実行します。期待する値になっているかどうかの確認は、組み込みのチェックポイント・メカニズムまたはカスタム C# コードで実行できます。

SOA とは

近年、アウトソーシングや高度なビジネスのニーズにより、複数の企業が協力し、情報を共有することが必要になりました。さらに、企業の合併と買収も増加していることから、各企業は、データの共有に苦勞するようになってきました。それぞれ独自のコンピュータ・システムを持つ 2 つの企業が合併する場合、合併後にデータを共有するために、大変な時間とコストがかかることがあります。

このような問題のために、B2B（ビジネス・ツー・ビジネス）通信を扱うテクノロジーがいくつかのベンダによって開発されました。B2B テクノロジーの例としては、RMI、COM、CORBA、EDI、Web サービスなどが挙げられます。これらのテクノロジーでは、システムを互いに接続できるようにするだけでなく、イントラネットのように相互のネットワークを許可する権限も処理します。

Web サービスは、インターネットを介してさまざまなプラットフォームで実行可能な自己完結型のアプリケーションです。これは、基本言語として XML と SOAP (Simple Object Access Protocol) を使用しており、開発者が扱いやすいソリューションとなっています。Web サービスは、標準化された一連のルールと仕様に基づいているため、ほかのテクノロジーより容易に移植できます。

SOA (サービス指向アーキテクチャ) は、複数のソフトウェア・サービスが相互に作用可能なアーキテクチャ・スタイルです。サービスは、サービス・プロバイダによって、コンシューマが指定するように実行される作業単位です。SOA では、複数のサービスが相互に依存することなく、相互に作用することが求められます。各サービスは自立的で結合が緩く、相互に認識することは必要ですが、依存性は求められません。

SOA システムは、主として Web サービスをベースにしています。Web サービスでは、SOAP プロトコルを使用して、クライアントが要求を送信し、Web サーバが応答を提供します。API テスト用 HP Unified Functional Testing では、Web サービスの動作チェックを自動化して実行できます。

SOA テストを自動化する理由

自動化された SOA テストは、アプリケーションのアップグレード時や新しいサービスのデプロイメント時におけるリスクを低減するための製品とプロセスを活用するための 1 つの手段です。その中心である自動化されたテストの目的は、実稼動負荷をデプロイメント前のシステムに適用しながら、同時にシステム・パフォーマンスとエンドユーザのエクスペリエンスを測定することです。パフォーマンス・テストを適切に構築すると、次のような質問に答えることができます。

- ▶ サービスの応答時間は、対象とするユーザにとって十分に短いか?
- ▶ サーバは、正しい値を応答するか?
- ▶ サービスは、例外や不正な値をどのように扱うか?
- ▶ サービスは、想定内および想定外のユーザ負荷の下で安定しているか?

これらの質問に答えることで、テストをより効果的に設計できます。効果的な自動テスト・プロセスにより、公開の決定を情報に基づいて行い、システム停止時間を短縮し、可用性の問題を防ぐことができます。

用語について

SOA テストを使用していると、次のような用語を目にすることがあります。

HTTP	ハイパーテキスト転送プロトコル : World Wide Web 経由での情報の転送や提供に使用される通信プロトコル。ユーザは、HTTP を利用して、HTML ページの公開と取得を行います。
JMS	Java Message Service : Java ベースのメッセージ指向ミドルウェア API で、複数のクライアント間でメッセージを送信するために使用します。
REST	Representational State Transfer : 分散システム向けのソフトウェア・アーキテクチャ・スタイル。
SOAP	Simple Object Access Protocol または Service Oriented Architecture Protocol : HTTP または JMS で送信される XML を使用して、型付きの構造化情報を分散環境内のピア間で交換するためのプロトコル。SOAP を使用することで、分散されたコンポーネントをシリアライズして転送用の XML ドキュメントにしたり、それが宛先に届いたときにデシリアライズすることができます。これにより、ベース・テクノロジーが異なるコンポーネント間の相互運用性が促進されます。
テスト	ユーザが実行するステップは、 テスト 内に記述されます。テストは、アプリケーションを使用するユーザの実際のアクションをエミュレートします。
UDDI	Universal Description, Discovery and Integration : プラットフォームに依存しないオンライン・レジストリで、世界中のビジネスをリストします。公開 Web サービス用のデータベースとしてよく使用されます。
Web サービス	標準化された Web ベース・アプリケーションで、インターネット・プロトコル上で XML, SOAP, WSDL, UDDI 標準を使用します。XML はデータのタグ付けに使用し、SOAP はデータの転送に使用し、WSDL はサービスを記述し、UDDI はそのサービスをリストします。

WSDL	Web サービス記述言語 : Web サービスを記述するための XML ベース言語です。WSDL ドキュメントは、実装するために必要な、Web サービスに関する基本情報（ポート、操作など）を提供します。
WS-I	Web Service Interoperability : Web Service Interoperability 組織が Web サービス間の互換性を促進するために作成した標準です。
XML	拡張マークアップ言語 は、汎用のマークアップ言語です。これが拡張（可能）と呼ばれるのは、ユーザ独自のタグを定義できるためです。この言語により、構造化データを異なる情報システム間で、特にインターネット経由で共有できます。XML は、データのシリアライズとドキュメントのエンコードの両方に使用されます。
XSD	XML スキーマ定義 ファイルには、XML ドキュメントの階層構造を正式に定義する一連のルールが含まれます。このルールと制約に準拠しないドキュメントは、パーサとプロセッサで有効とはみなされません。 WSDL ドキュメントは外部スキーマを参照するか、組み込みのスキーマを持つことができます。

このチュートリアルサンプル・アプリケーション

このチュートリアルは、この製品に付属するサンプル・アプリケーション (**API Flights**) をベースにしています。このアプリケーションは、Web サービスと REST サービスの両方として使用できます。

サンプル・アプリケーションは、フライト予約データベースとともに使用します。特定の行き先のフライトの検索、顧客注文の作成、予約の更新、予約の削除を実行できます。

サービスのメソッドと操作の詳細については、サンプル・アプリケーションのコマンド・プロンプト・ウィンドウで「help」と入力してください。

注: データ・ソースに対して Excel 機能を有効にするため、使用するマシンに Microsoft Excel をインストールしておくことをお勧めします。

アプリケーションを起動する方法

最初の手順として、サンプルのフライト・アプリケーションを開始し、テストで使用できるようにします。

HP Flights サービスを開始するには、次の手順を実行します。

- 1 管理者権限があることを確認してください。Windows でサンプルの HP Flights サービスを実行する際に必要となります。
- 2 **[スタート] > [(すべての) プログラム] > [HP Software] > [HP Unified Functional Testing] > [Sample Applications] > [Flight API]** を選択します。コマンド・ウィンドウが開いて、アプリケーションを使用できることが示されます。
- 3 標準設定のポート 24240 を使用できないというメッセージがウィンドウに表示された場合は、**<インストール・ディレクトリ>SampleApplication\HPFlights_Service.exe.config** ファイルをテキスト・エディタで編集します。**appSettings** セクションの 24240 ポート・キーを有効な値に変更してください。
- 4 サンプル・アプリケーションのコマンド・ウィンドウを最小化します。コマンド・ウィンドウは閉じないでください。閉じると、サービスが停止します。

これからの作業

アプリケーションを起動できたので、API Flights アプリケーションを使用して、ヘッドレス・アプリケーション用のテストの作成に取りかかることができます。ここからのレッスンでは、基本的なアクティビティ、Web サービス、REST サービスのテストを作成するプロセスを順を追って説明します。

レッスン 2

単純なテストの作成

このレッスンでは、単純なアクションがあるテストの作成手順について説明します。

このレッスンの内容は次のとおりです。

- ▶ 新しいテストを作成する方法
- ▶ Unified Functional Testing の表示枠を使用する方法
- ▶ テスト・ステップを作成する方法
- ▶ ステップをデータ駆動する方法
- ▶ テスト・ステップを接続する方法
- ▶ 複数のソースからのデータをマップする方法

新しいテストを作成する方法

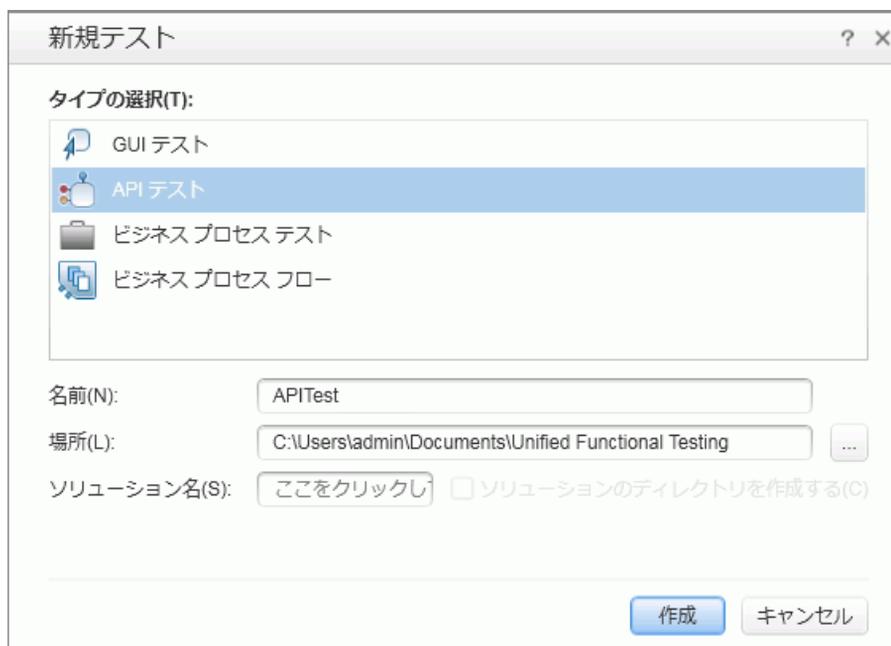
テストを作成し実行する中心的なコンソールはキャンバスです。

1 HP UFT を開きます。

[スタート] > [(すべての) プログラム] > [HP Software] > [HP Unified Functional Testing] > [Sample Applications] > [HP Unified Functional Testing] を選択します。開始ページが開きます。

2 新しいテストを作成します。

[ファイル] > [新規] > [テスト] をクリックします。[新規テスト] ダイアログ・ボックスが開きます。[API テスト] タイプを選択します。HP LoadRunner がインストールされているマシンでは、[API 負荷テスト] タイプも表示されます。



3 新しいソリューションを生成します。

[名前] ボックスの標準設定の名前を **BasicTest** に変更し, [作成] をクリックします。空のテストが開いて, キャンバスに [開始], [テスト フロー], [終了] の各セクションが表示されます。

[テスト フロー] は, テストする機能のアクティビティが含まれるテスト・セクションです。[開始] セクションは, テスト前に初期化する項目 (テスト変数など) の定義に最適です。

Unified Functional Testing の表示枠を使用する方法

Unified Functional Testing インタフェースのほとんどの表示枠は, ドッキング可能なフローティング・タイプのウィンドウです。標準設定の表示枠を元々の位置に表示するには, [表示] > [ウィンドウ レイアウトのリセット] を選択します。

主要な表示枠を次に示します。

- ▶ **[ソリューション エクスプローラ] 表示枠** : (左) 現在のソリューションのすべてのテストとアクションのツリー階層で, 参照, フロー, イベントも表示されます。
- ▶ **[ツールボックス] 表示枠** : (左) テスト・ステップとして追加可能な組み込みのアクティビティとインポートされたアクティビティが集められています。この表示枠から, アクティビティをキャンバスにドラッグできます。
- ▶ **キャンバス** : (中央) テスト・ステップを編成する作業領域です。
- ▶ **[データ] 表示枠** : (下) テストで使用できるデータ・ソースのツリー階層です。データ・ソースには, インポートされた Excel および XML ファイル, データベース・テーブル, 手動で定義したテーブルなどがあります。
- ▶ **[出力] 表示枠** : (下) 情報を伝える領域で, テストの実行とステータスに関する情報を提供します。

テスト・ステップを作成する方法

テスト・ステップは, [ツールボックス] 表示枠のアクティビティをキャンバスにドラッグして作成します。

ここでは, 単純なテスト・ステップを作成して, ツールボックスや各種表示枠の使用方法を学びます。

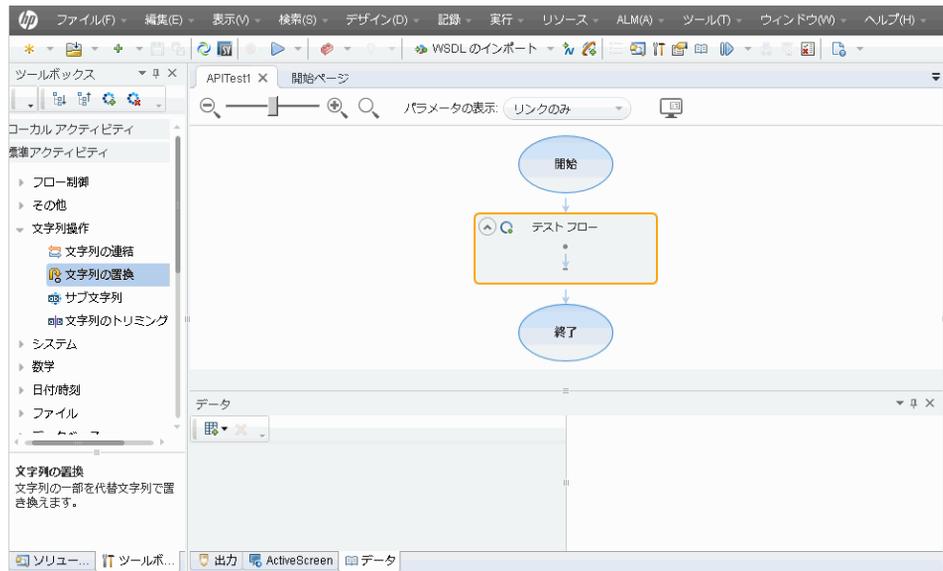
単純な文字列置換テスト・ステップの作成：

1 [文字列の置換] アクティビティを見つけます。

左の表示枠で、[ツールボックス] タブをクリックして、[ツールボックス] 表示枠を表示します。[文字列操作] カテゴリを展開し、[文字列の置換] を選択します。

2 ステップを作成します。

[文字列の置換] アクティビティをキャンバスにドラッグし、**テスト・フロー**にドロップします。このアクティビティは、テキストを特定の文字列の中から検索し、そのテキストを新しいテキストで置き換えます。この操作の代わりに、ツールボックスのアクティビティをダブルクリックして、キャンバスに追加することもできます。



3 [一般] ビューでステップの表示名を変更します。

[表示] > [プロパティ] を選択します。キャンバスで [文字列の置換] ステップを選択し、[プロパティ] 表示枠で [一般] タブをクリックします。[名前] 行に「テキストの変更」と入力し、ENTER キーを押します。これで、キャンバス内のステップ名が変更されます。

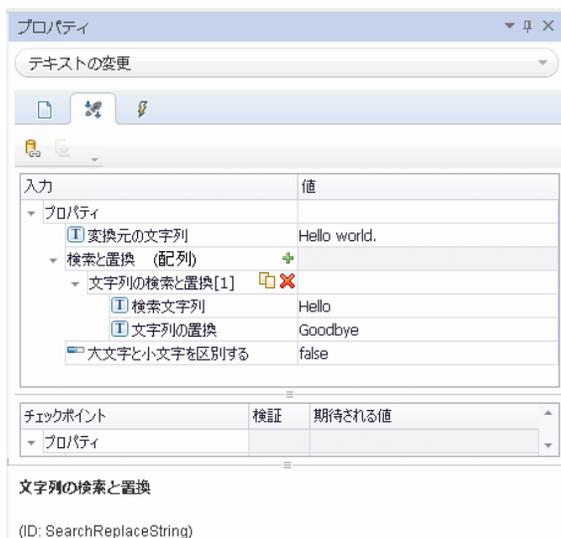


4 入力プロパティを設定します。



[プロパティ]表示枠で[入力/チェックポイント]タブを選択します。次の値を入力します。

- ▶ **変換元の文字列** : Hello world.
- ▶ **検索文字列** : Hello
- ▶ **置換文字列** : Goodbye
- ▶ **大文字と小文字を区別する** : false



5 テストを実行します。



[実行] ボタンをクリックするか F5 キーを押して、[テストの実行] ダイアログ・ボックスを開きます。[オプション] をクリックして、ダイアログ・ボックスを展開します。[一時実行結果フォルダ] オプションを選択します。[実行] をクリックして、テストをコンパイルし実行します。

6 結果を表示します。

Run Results Viewer が開きます。

[表示] > [すべて展開] を選択するか、[すべて展開] ツールバー・ボタンをクリックします。[テキストの変更] ノードをクリックします。変換元の文字列と置換する文字列を確認し、結果文字列の **Goodbye world.** に着目してください。これは、実際に期待した文字列です。したがって、テストはパスしました。

結果の確認が終了したら、Run Results Viewer を閉じます。

The screenshot shows the HP Run Results Viewer window. The left pane shows a tree view of test steps, with 'テキストの変更' selected. The main area displays the details for this step, including a table of results and a properties table.

オブジェクト	詳細	結果	時間
テキストの変更		完了	2013/03/12 - 11:02:49

名前	値
タイプ	HP.ST.Ext.BasicActivities.ReplaceStringActivity
ステップ ID	ReplaceStringActivity4
変換元の文字列	'Hello world.'
検索文字列 1	'Hello'
文字列 1 の置換	'Goodbye'
文字列 1 の置換の回数	1
メッセージ	1 文字列が置換されました
大文字と小文字を区別する	False
結果	'Goodbye world.'
名前	'テキストの変更'
コメント	"
ステータス	完了

7 チェックポイントを設定します。

1つ前の手順では、出力を手動で表示して、結果が期待値に一致しているかどうかを確認しました。チェックポイントを使用すると、結果を手動でチェックしなくても、アクションが成功したかどうかを確認できます。チェックポイントは、テストを検証する手段です（成功したか失敗したかは、そのチェックポイントによって決定されます）。

[プロパティ] 表示枠（右の表示枠）に戻り、[入力/チェックポイント] タブが表示されていることを確認します。表示枠の下部にある [チェックポイント] セクションをクリックし、[結果] 行のチェック・ボックスを選択して、チェックポイントを有効にします。[期待値] カラムに、期待する文字列「**Goodbye world.**」を入力します。

テストを再度実行します。Run Results Viewer でノードを展開し、チェックマークに着目します。このマークは、結果と期待値が一致したため、チェックポイントがパスしたことを示します。

結果の確認が終了したら、Run Results Viewer を閉じます。

テスト・ステップを接続する方法

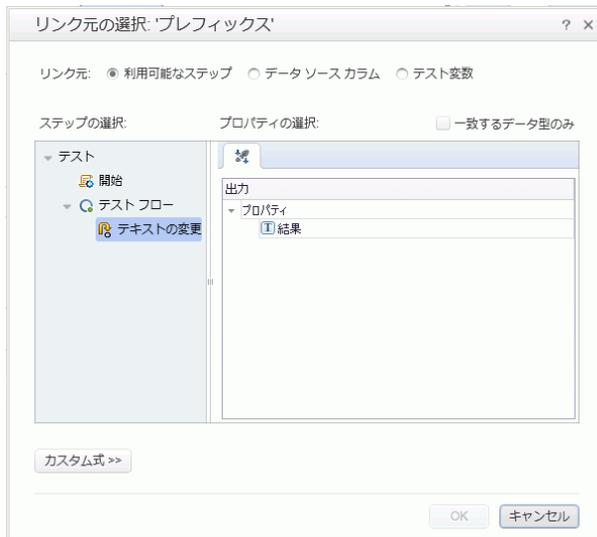
ここでは、あるステップの出力を別のステップの入力として使用します。

1 文字列の連結ステップを追加します。

[**ツールボックス**] 表示枠で、[**文字列操作**] カテゴリから [**文字列の連結**] を選択します。このアクティビティをキャンバスにドラッグし、テスト・フローの**テキストの変更**ステップの下にドロップします。このアクティビティは、2つの文字列を連結します。

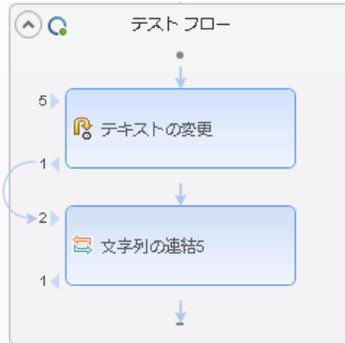
2 プレフィックスを設定します。

キャンバスで [**文字列の連結 (x)**] ステップを選択します。[**プロパティ**] 表示枠で [**入力/チェックポイント**] タブをクリックします。上部の [**入力**] セクションで、[**プレフィックス**] 行の [**値**] セルにマウスを移動します。[**データ ソースにリンク**] ボタンをクリックします。[**リンク元の選択**] ダイアログ・ボックスが開きます。



3 ステップをリンクします。

[リンク元の選択] ダイアログ・ボックスで、[利用可能なステップ] オプションを選択します。[テスト フロー] > [テキストの変更] ノードを選択します。右の表示枠で、[結果] ノードをダブルクリックします。これで、データが**テキストの変更**から**文字列の連結**に移動することがキャンバスに反映されます。



4 サフィックスを設定します。

[プロパティ] 表示枠で、[サフィックス] プロパティの [値] フィールドに「Welcome to the Basic Test.」と入力します。

入力	値
プロパティ	
プレフィックス	{Step.OutputProperties.Repl
サフィックス	Welcome to the Basic Test

5 テストを実行します。



[実行] ボタンをクリックするか F5 キーを押して、テストを実行します。

6 レポートを表示します。

実行結果ツリーを展開し、[文字列の連結] アクティビティ・ノードを選択します。レポートに、連結された文字列の結果が表示されます。**Goodbye World.Welcome to the Basic Test.**

結果の確認が終了したら、Run Results Viewer を閉じます。

複数のソースからのデータをマップする方法

[リンク元の選択] ダイアログ・ボックスを使用すると、1つまたは複数のデータ・ソース（**利用可能なステップ**、**データ ソース カラム**、**テスト変数**）をリンクして、入力値を指定できます。前の項では、1つの値について**利用可能なステップ**・ソースを使用し、もう1つの値についてはデータを手入力しました。

カスタム式を作成すると、複数のデータ・ソースをプロパティ値として使用できます。ここでは、[リンク元の選択] ダイアログ・ボックスを使用して、手入力値と [**利用可能なステップ**] オプションの自動値の両方を使用する**サフィックス**・プロパティの式を作成します。

1 プレフィックスを設定します。



キャンバスで [**文字列の連結**] ステップを選択します。[プロパティ] 表示枠で [**入力/チェックポイント**] ビューを開きます。[**プレフィックス**] 行の [**値**] セルをクリックし、[**X**] をクリックして内容をクリアします。新しいプレフィックス「Hello world.」を入力します。



2 [リンク元の選択] ダイアログ・ボックスを開きます。



[**サフィックス**] 行の [**値**] セルをクリックし、[**X**] をクリックして内容をクリアします。[**データ ソースにリンク**] ボタンをクリックします。[リンク元の選択] ダイアログ・ボックスが開きます。



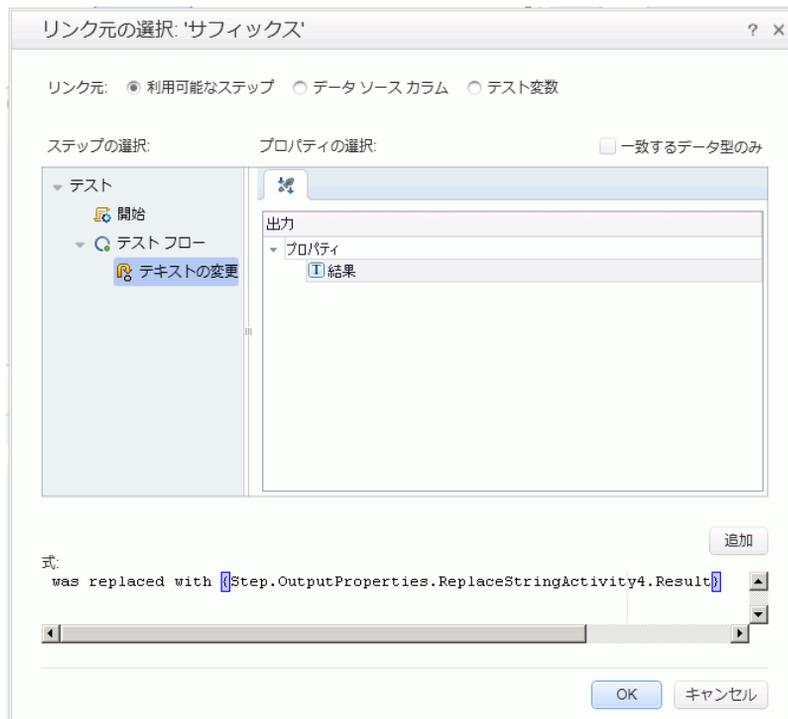
3 サフィックスを編集します。

[リンク元の選択] ダイアログ・ボックスで、[**カスタム式**] ボタンをクリックして展開します。[**式**] ボックスに、「was replaced with」と入力します（読みやすいようにフレーズの前後にスペースを追加します）。



4 ソースをもう1つ追加します。

[**利用可能なステップ**] オプションを選択し、左の表示枠で [**テキストの変更**] ノードを選択します。右の表示枠で [**結果**] ノードを選択し、[**追加**] をクリックします。[**式**] ボックスに両方のソースが表示されます。



5 テストを実行し、レポートを表示します。



[実行] ボタンをクリックして、テストを実行します。結果ツリーを展開し、[文字列の連結] ノードを選択します。レポートに、連結された文字列の結果が表示されます。

Captured Data	
ステップのプロパティ	
名前	値
タイプ	HP.ST.Ext.BasicActivities.ConcatenateStringsActivity
ステップ ID	ConcatenateStringsActivity5
メッセージ	文字列は正常に連結されました
プレフィックス	'Hello world.'
サフィックス	' was replaced with Goodbye world.'
結果	'Hello world. was replaced with Goodbye world.'
名前	'文字列の連結5'
コメント	"
ステータス	完了

6 Run Results Viewer ウィンドウを閉じます。

ステップをデータ駆動する方法

データ駆動とは、データ・ソース（Excel または XML ファイル、データベース、ローカル・テーブルなど）のデータをテスト・ステップに割り当てる処理です。データ駆動の目的は、同じビジネス・プロセスを異なるデータで実行することです。これにより、データ・テーブルを変更するだけで、アプリケーションをさまざまな状況でチェックできます。

テスト・ステップをデータ駆動するには、次の手順を実行します

1 入力引数をデータ駆動します。



キャンバスで [テキストの変更] ステップを選択します。[プロパティ] 表示枠で [入力 / チェックポイント] ビューを開き、[ステップ全体のデータ駆動] ボタンをクリックします。[データ駆動] ダイアログ・ボックスが開きます。

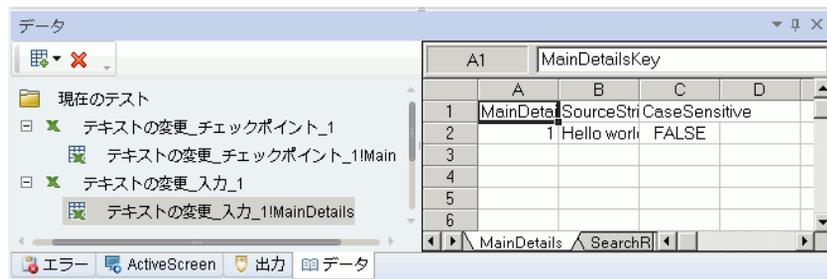
2 データ・プロバイダを指定します。

[データ駆動] ダイアログ・ボックスで：

- a [データ プロバイダ] タイプを [Excel] に設定します。
- b [入力プロパティとチェックポイントの両方] のデータ駆動を有効にします。
- c ['テスト フロー' を新しいデータ ソースを使用して ForEach ループとして構成します] オプションをクリアします。これで、データ行の数に基づいてテスト・フローが繰り返されます。反復の回数は後の手順で手動で設定します。
- d [OK] をクリックして、[データ駆動] ダイアログ・ボックスを閉じます。
- e ポップアップ・メッセージを受け入れます。データ駆動のメカニズムによって、定数値が新しい式 {DataSource.テキストの変更_入力!MainDetails.SourceString} で置換されます。

3 [データ] 表示枠を表示します。

[データ] 表示枠が開いていることを確認します。開いていない場合は、[表示] > [データ] を選択します。[テキストの変更_入力] ノードを展開し、[テキストの変更_入力!MainDetails] ノードを選択します。[データ] 表示枠に、各入力プロパティの列と、前に入力したプロパティに対応する値 (Hello World. および FALSE) の 1 行が表示されます (Excel のないインストール環境では空のチェック・ボックスが表示されます)。



4 新規データを追加します。

[テキストの変更_入力!MainDetails] テーブルに、さらに2行追加します。テキストは、句読点も含めて正確にコピーしてください。

MainDetailsKey	SourceString	CaseSensitive
1	Hello world.	FALSE
2	I like eating broccoli.	TRUE
3	The product version is 11.	FALSE

5 新しい「検索と置換」データを追加します。

[テキストの変更_入力!SearchReplaceString] ノードをクリックし、テーブルに2行追加します。テキストは、句読点も含めて正確にコピーしてください。

MainDetailsKey	Key	Value	CaseSensitive
1	Hello	Goodbye	FALSE
2	broccoli	ice cream	TRUE
3	11	12	FALSE

6 チェックポイントの値を追加します。

[テキストの変更_チェックポイント] ノードを展開し、[テキストの変更_チェックポイント!MainDetails] ノードを選択します。このコラムに、値を次のように追加します。

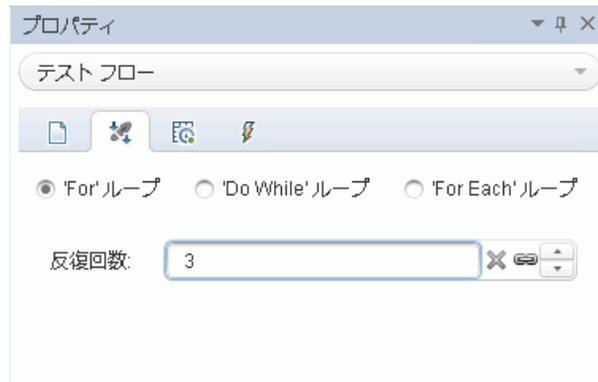
注： エラーを発生させるために、3 行目では感嘆符 (!) を故意に追加します。

Result
Goodbye world.
I like eating ice cream.
The product version is 12!

7 反復回数を設定します。

反復回数は、ステップを繰り返す回数です。これは、テーブルのデータ行数に合わせて 3 に設定します。

キャンバスに戻り、テスト・フロー・フレーム内をクリックします。このとき、テスト・ステップ内をクリックしないようにしてください。[プロパティ] 表示枠で [入力/チェックポイント] ビューを開きます。[For ループ] を選択し、[反復回数] を 3 に設定します。



8 テストを実行し、レポートを表示します。



[実行] ボタンをクリックするか F5 キーを押して、テストをコンパイルし実行します。テストは、テーブルの3行のデータを使用して、3回実行されます。

テストの実行後、Run Results Viewer が開きます。[テスト フロー] ノードを展開し、赤の **X** が付けられた行までドリルダウンします（このマークは失敗したチェックポイントを示します）。チェックポイントが失敗したのは、期待する結果に、ソース文字列にはない感嘆符が含まれていたためです。

9 エラーを修正し、テストを再実行します。

[データ] 表示枠で、データの [テキストの変更_チェックポイント!MainDetails] ノードを修正します。3行目の [Result] カラムのチェックポイントについて、感嘆符をピリオドに変更します。

スクリプトを再度実行し、レポートにエラーがないことを確認します。

これからの作業

これで、単純なテスト・ステップを作成できるようになったので、Web サービスを使用してステップを作成できます。ここからのレッスンでは、WSDL をインポートし Web サービス・テストを作成するプロセスを順を追って説明します。

レッスン 3

Web サービスのテスト

UFT API テストを使用して、WSDL ベース Web サービスのテストを作成できます。

このレッスンの内容は次のとおりです。

- ▶ Web サービスをインポートする方法
- ▶ Web サービス・テストを作成する方法
- ▶ テストにデータを組み入れる方法
- ▶ 複数のデータ・ソースとカスタム・コードを使用する方法

Web サービスをインポートする方法

WSDL ファイルでは、Web サービスの操作が定義されます。WSDL ファイルを使用するには、そのファイルをテストにインポートします。ここでは、サンプル・アプリケーションの WSDL ファイルをインポートする方法を示します。

1 サンプル・フライト・アプリケーションを開始します。

フライト・アプリケーション・サービスが使用できることを、13 ページ「アプリケーションを起動する方法」の説明に従って確認します。

2 新しいソリューションを作成します。

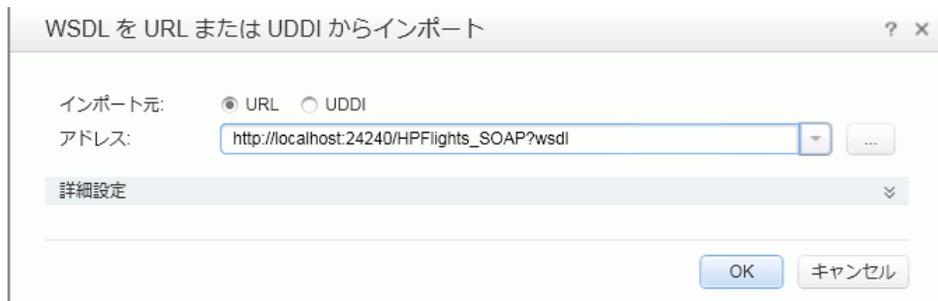
[ファイル] > [新規] > [テスト] を選択し、新しい API テストの名前として **WebServiceTest** を指定します。[作成] をクリックします。

3 [サービスのインポート] ダイアログ・ボックスを開きます。

ツールバーで [WSDL のインポート] > [WSDL を URL または UDDI からインポート] を選択します。

4 インポートのソースを指定します。

[URL] オプションを選択し、場所 **http://localhost:24240/HPFlights_SOAP?wsdl** を指定します。[OK] をクリックします。



5 サービスの操作を表示します。

インポートによって、[ツールボックス] の [Web サービス] カテゴリの下に、Web サービス操作の新しいブランチが作成されています。ノードを展開して、操作を表示します。



Web サービス・テストを作成する方法

ここでは、**HPFlights** Web サービスを使用して、航空券の申し込みを新規に作成します。

航空券の申し込みを作成するには、まず利用可能なフライトを知っておく必要があります。最初に、**GetFlights** ステップを実行して、目的地へのフライトをすべて検索します。その次に、返された最初のフライト・ナンバーを **CreateFlightOrder** ステップの入力として使用します。

1 GetFlights ステップを作成します。

[Web サービス] > [HPFlights_Service] ノードを展開し、**GetFlights** アクティビティをテスト・フローにドラッグします。

2 DepartureCity と ArrivalCity の値を割り当てます。



[入力/チェックポイント] ビューを開いて、[Body] > [GetFlights] ノードを展開します。都市を選択するために、行内の矢印をクリックして、ドロップダウン・リストを開きます。[DepartureCity] として Denver を選択し、[ArrivalCity] として Los Angeles を選択します。

入力	値
Envelope	
Header	
Any (配列)	
Body	
GetFlights	
DepartureCity	Denver
ArrivalCity	Los Angeles

チェックポイント	検証	期待される値
Envelope	<input type="checkbox"/>	=
Header	<input type="checkbox"/>	=
Any (配列)	<input type="checkbox"/>	=
Body	<input type="checkbox"/>	=
GetFlightsResponse	<input type="checkbox"/>	=

XML XPath

要求をサービスに送信 構造の検証 WS-I の検証

3 CreateFlightOrder ステップを作成します。

ツールボックスの **CreateFlightOrder** アクティビティを、テスト・フローの **GetFlights** ステップの下にドラッグします。

4 CreateFlightOrder ステップの値を設定します。



[入力/チェックポイント] ビューで, [Body] > [CreateFlightOrder] > [FlightOrder] ノードを展開し, 航空券の申し込みを作成するための値を設定します。

- ▶ **Class** : ドロップダウン・リストからクラス ([Business] など) を選択します。
- ▶ **CustomerName** : 任意の値
- ▶ **DepartureDate** : ドロップダウンを使用してカレンダーを開き, 本日より少なくとも 2 日先の日付を選択します。
- ▶ **FlightNumber** : 今は空欄にしておきます。これは, この先の手順で設定します。
- ▶ **NumberOfTickets** : スクローラで任意の値を設定します。

5 GetFlights の出力を CreateFlightOrder ステップにリンクします。



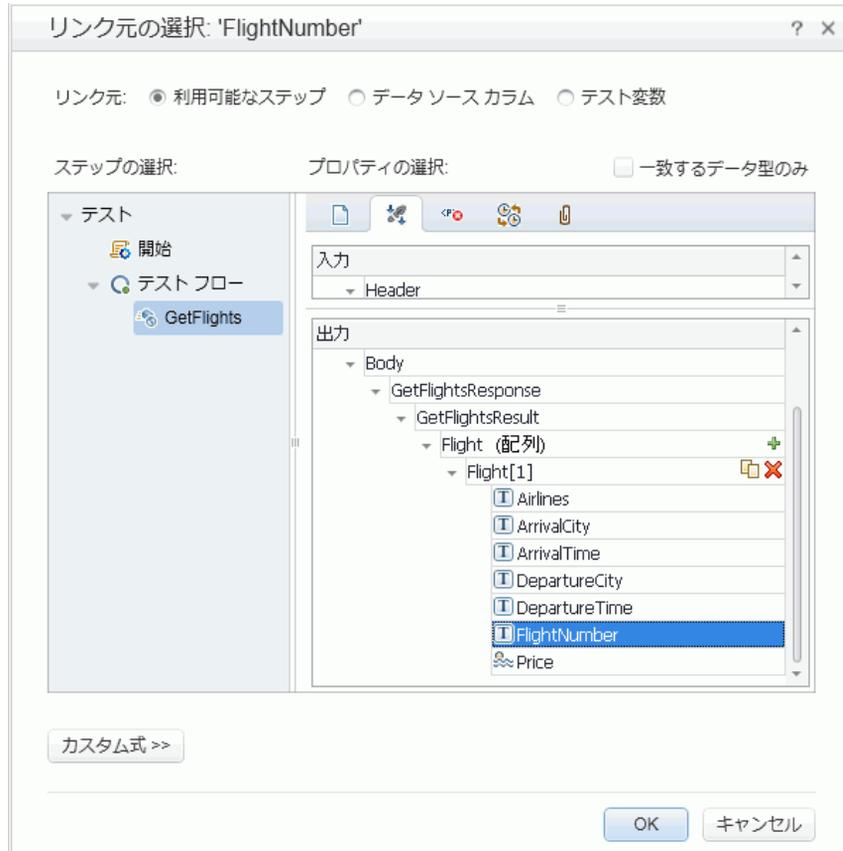
a [FlightNumber] 行の右隅にある [データ ソースにリンク] アイコンをクリックします。[リンク元の選択] ダイアログ・ボックスが開きます。

b [利用可能なステップ] を選択し, [GetFlights] ノードを選択します。

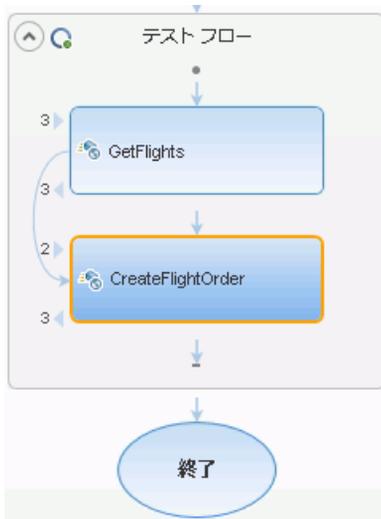
c 右の表示枠で, [入力/チェックポイント] ボタンを選択します。[出力] セクションで, [Body] ノードの下ノードをすべて展開します。**Flight (配列)** ノード行の [追加] ボタンをクリックして, **Flight[1]** 配列を作成します。



- d [Flight[1]] 配列を展開し、[FlightNumber] 要素を選択し、[OK] をクリックします。ターゲットのステップをループで囲むかどうか尋ねられます。[はい] を選択します。



キャンバスに2つのステップ間の接続が示されます。



6 反復回数をリセットします。

反復回数は、ステップを繰り返す回数です。キャンバスに戻り、テスト・フロー・フレーム内をクリックします。このとき、テスト・ステップ内をクリックしないようにしてください。[プロパティ] 表示枠の [入力] ビューを開きます。[For ループ] を選択し、[反復回数] を 1 に設定します。



7 テストを実行します。

[実行] ボタンをクリックします。[出力] タブでログを確認します。Run Results Viewer が自動的に開きます。



8 結果をチェックします。

左の表示枠で親ノードをクリックし、右クリック・メニューから **[すべて展開]** を選択します。**[CreateFlightOrder]** ノードをクリックします。**[キャプチャ データ]** 表示枠で、**[Web サービス コール HTTP スナップショット]** セクションまで下にスクロールし、**[応答]** 表示枠に着目します。要求の出力 (**OrderNumber** および **TotalPrice**) に着目します。**TotalPrice** の値を、この次の手順で使用できるようにクリップボードにコピーします。

要求	応答
HTTP ヘッダ SOAPAction: HP.SOAQ.SampleApp/HPFlights_Service/CreateF Content-Type: text/xml; charset=utf-8 Host: localhost:24240 Content-Length: 411 Expect: 100-continue Connection: Close	HTTP ヘッダ Connection: close Content-Length: 335 Content-Type: text/xml; charset=utf-8 Date: Wed, 13 Mar 2013 03:13:40 GMT Server: Microsoft-HTTPAPI/2.0
SOAP <pre><Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/"> <Body> <CreateFlightOrder xmlns="HP.SOAQ.SampleApp"> <FlightOrder> <Class>Business</Class> <CustomerName>Smith</CustomerName> <DepartureDate>2013-03-18T00:00:00+08:00</Departur <FlightNumber>1098</FlightNumber> <NumberOfTickets>1</NumberOfTickets> </FlightOrder> </CreateFlightOrder> </Body> </Envelope></pre>	SOAP <pre><?Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelop <s:Body> <CreateFlightOrderResponse xmlns="HP.SOAQ.SampleApp"> <CreateFlightOrderResult xmlns:i="http://www.w3.org/20 <OrderNumber>18</OrderNumber> <TotalPrice>118</TotalPrice> </CreateFlightOrderResult> </CreateFlightOrderResponse> </s:Body> </s:Envelope></pre>

ヒント: **[要求]** または **[応答]** リンクをクリックすると、別のブラウザで SOAP が開きます。

結果の確認が終了したら、Run Results Viewer を閉じます。

9 チェックポイントを設定します。

キャンバスで **CreatFlightOrder** ステップを選択します。**[プロパティ]** 表示枠の **[入力/チェックポイント]** ビューを開き、**[チェックポイント]** グリッドをクリックし、**[CreateFlightOrderResponse]** ノードを展開します。前の手順のクリックボードの内容を **[TotalPrice]** フィールドに貼り付けます。**[TotalPrice]** 行のチェック・ボックスを選択し、チェックポイントとして含めます。

10 テストを実行してチェックポイント結果を表示します。

テストを再度実行し、結果ツリーを展開します。**CreateFlightOrder** の [チェックポイント] ノードを選択します。レポートにチェックマークが表示され、期待される値と実際の値が示されます。期待した値がサーバから返されなかった場合、レポートは失敗を示します。

名前	結果	プロパティ	実際の結果	評価スタイル	期待値	詳細
"チェックポイント 1"	✓	構造的検証	...	
"チェックポイント 2"	✓	"Envelope[1]/Body[1]/CreateFlightOrderResponse[1]/CreateFlightOrderResult[1]/TotalPrice[1]"	"116"	=	"116"	...

結果の確認が終了したら、Run Results Viewer を閉じます。

テストにデータを組み入れる方法

ここでは、既存のソースのデータを組み込む方法と、テストをデータ駆動する方法について学習します。テストをデータ駆動すると、値を編集できるデータ・テーブルが [データ] 表示枠自動的に作成されます。

1 サンプル・データのインポート

[データ] 表示枠 (UFT ウィンドウの下部) で、[新規データ ソース] > [Excel] を選択します。[新規 Excel データ ソース] の追加ダイアログ・ボックスが開きます。

- a サンプル・アプリケーションの Excel ファイル **SampleAppData.xlsx** を指定します。これは、<インストール・ディレクトリ>/**SampleApplication** フォルダにあります。標準設定では、このフォルダは **C:\Program Files (x86)\HP\Unified Functional Testing\samples\flight_service** です。
- b サンプル・ファイルにはヘッダ行が含まれているため、[Excel ファイルにヘッダ行が含まれています] オプションを有効にします。

- c [データ ソース名] として「WS_Flights」を入力します。
- d インポートのモードとして [参照データ ソース] を選択します。これにより、元の場所にある Excel ファイルにリンクされ、データに変更があった場合も、使用するデータ・ソースは最新のものになります。
- e [OK] をクリックします。

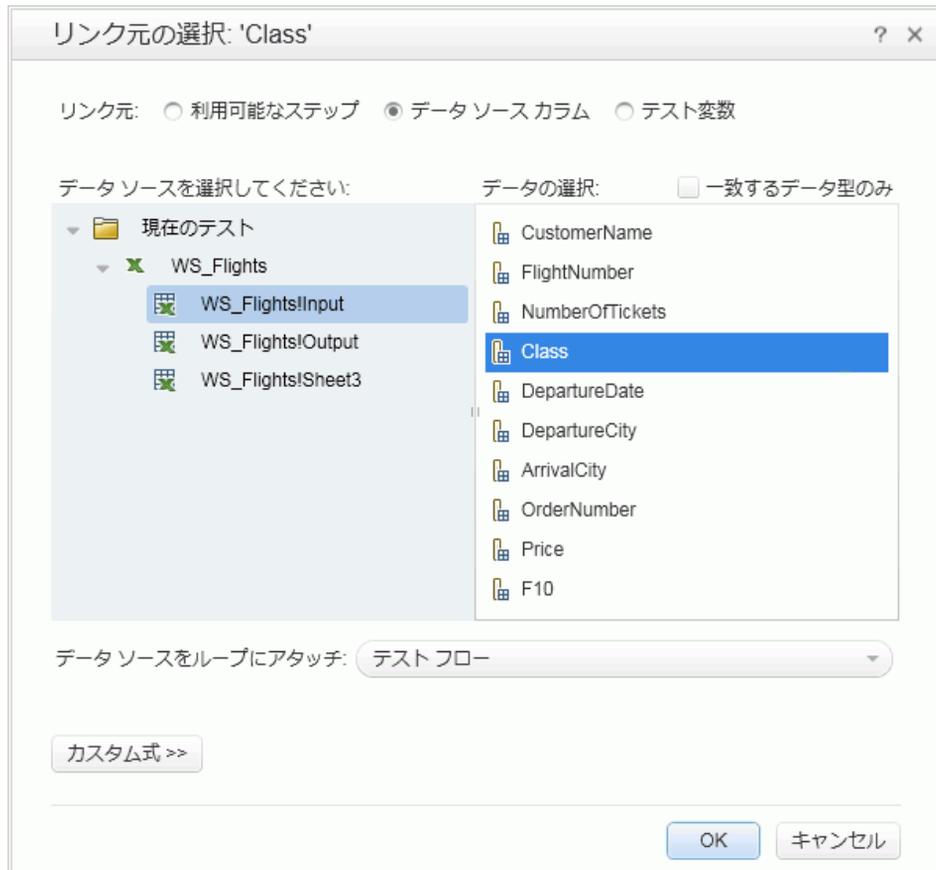
2 [リンク データの選択] ダイアログ・ボックスを開きます。

キャンパスで **CreateFlightOrder** ステップを選択し、[入力/チェックポイント] ビューを開きます。[入力] セクションですべてのノードを展開し、[Class] 行を選択します。[データ ソースにリンク] アイコンをクリックします。[リンク元の選択] ダイアログ・ボックスが開きます。



3 データ・ソースの値を選択します。

[データ ソース カラム] オプションを選択します。



4 サンプルの Excel データを使用します。

[WS_Flights!Input] ノードを選択し、右の表示枠で [Class] をクリックします。[OK] をクリックします。これにより、テストの実行中にサンプルデータのこのカラムを参照するように、テストに対して指示が出されます。

これを、ほかの入力パラメータ (**CustomerName**, **DepartureDate**, **FlightNumber**, **NumberOfTickets**) についても繰り返します。

5 チェックポイントを無効にします。

[入力/チェックポイント] ビューで [チェックポイント] グリッドをクリックします。
 [TotalPrice] プロパティのチェック・ボックスをクリアして、チェックポイントから除外します。

6 ナビゲーション設定を構成します。

ナビゲーションの設定により、データ・ソースのデータをどのように使用するかを示すことができます。開始する行、進む行数、次の値セットを求めて移動する方向を指定できます。また、データ・テーブルの最後に達したときの動作を、「最初に戻る」または「最後の行を引き続き使用する」から指定することもできます。

a キャンバスで、**テスト・フロー**内をクリックします。このとき、ステップ内をクリックしないようにしてください。



b [プロパティ] 表示枠で、[**データ ソース**] ビュー・ボタンをクリックします。

c [**WS_Flights!Input**] ノードを選択し、[**編集**] をクリックして、[データ ナビゲーション] ダイアログ・ボックスを開きます。

- d** データ・ナビゲーションの詳細を、**開始位置**：最初の行、**移動**：移動単位：3 行 進む、**終了位置**：最後の行、**最後の行への到達時**：折り返すに設定します。

データナビゲーション

開始

開始位置: 最初の行

行: 1

移動

移動単位: 3 行 進む

終了

終了位置: 最後の行

行: 7

最終行への到達時

アクション: 折り返す

OK キャンセル

- e** [OK] をクリックします。

7 テストを実行し、結果を表示します。



[実行] ボタンをクリックし、[出力] ウィンドウで結果を確認します。Run Results Viewer が自動的に開きます。結果ツリーを展開し、**CreateFlightOrder** ステップを選択します。[キャプチャ データ] タブ内を下にスクロールし、SOAP 要求 (左の表示枠) の Excel ファイル・データと、SOAP 応答 (右の表示枠) の結果に着目します。

SOAP	SOAP
<pre><Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/"> <Body> <CreateFlightOrder xmlns="HP.SOAQ.SampleApp"> <FlightOrder> <Class>Business</Class> <CustomerName>John Freeman</CustomerName> <DepartureDate>2015-10-10T00:00:00</DepartureDate> <FlightNumber>1042</FlightNumber> <NumberOfTickets>1</NumberOfTickets> </FlightOrder> </CreateFlightOrder> </Body> </Envelope></pre>	<pre><s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelop <s:Body> <CreateFlightOrderResponse xmlns="HP.SOAQ.SampleApp"> <CreateFlightOrderResult xmlns:i="http://www.w3.org/20 <OrderNumber>19</OrderNumber> <TotalPrice>125</TotalPrice> </CreateFlightOrderResult> </CreateFlightOrderResponse> </s:Body> </s:Envelope></pre>

結果の確認が終了したら、Run Results Viewer を閉じます。

複数のデータ・ソースとカスタム・コードを使用する方法

ここでは、複数のデータ・ソースを使用し、カスタム・コード・ステップで情報をレポートに送信して、データを定義する方法について説明します。

1 新しいテストを作成します。

新しいテストを **WebServicesCustom** という名前で作成し、32 ページ「Web サービスをインポートする方法」の説明に従って、HP Flights Services WSDL をインポートします。

2 テスト・ステップを作成します。

キャンバスにアクティビティを順番にドラッグします。[Web サービス] フォルダから **GetFlights** と **CreateFlightOrder** をドラッグします。[その他] フォルダから **カスタム・コード** をドラッグします。

3 データ・ソースを追加します。

[データ] 表示枠で、[新規データ ソース] > [Excel] を選択します。[新規 Excel データ ソース] の追加ダイアログ・ボックスで、次の手順を実行します。

- a サンプル・アプリケーション Excel ファイルを指定します。標準設定では、このフォルダは **C:\Program Files (x86)\HP\Unified Functional Testingsamples\flight_service** です。
- b [Excel ファイルにヘッダ行が含まれています] チェック・ボックスを選択します。

c [データ ソース名] として「WS_Flights」を入力します。

d [参照データ ソース] モードを選択します。

4 GetFlights の値を割り当てます。



キャンバスで **GetFlights** ステップを選択し、[プロパティ] 表示枠で [入力/チェックポイント] ビューを開きます。[入力] セクションで、**DepartureCity=Denver** と **ArrivalCity=Los Angeles** を選択します。

5 CreateFlightOrder の値を割り当てます。



キャンバスで **CreateFlightOrder** アクティビティを選択し、[入力/チェックポイント] ビューを開きます。[Body] > [FlightOrder] ノードを展開し、入力プロパティを次のように設定します。

▶ **Class** : Economy



▶ **CustomerName** : [CustomerName] 行の右隅にある [データ ソースにリンク] ボタンをクリックします。[リンク元の選択] ダイアログ・ボックスが開きます。[データ ソース カラム] を選択し、ツリーを展開して [WS_Flights!Input] ノードを表示します。右の表示枠で、[CustomerName] パラメータを選択します。[OK] をクリックします。

▶ **DepartureDate** : 書式が YYYY-MM-DDTHH:MM:SS の日付です (例 : 2015-02-18T00:00:00)。カレンダーを開くにはドロップダウン矢印を使用します。日付は、現在の日付より少なくとも2日先である必要があります。

▶ **NumberofTickets** : 3

▶ **FlightNumber** : 1つ前のステップからリンクします。

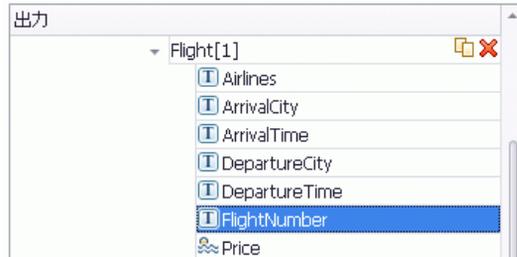


a [FlightNumber] 行の右隅にある [データ ソースにリンク] ボタンをクリックします。

b [データ ソースにリンク] ダイアログ・ボックスで [利用可能なステップ] を選択し、[テスト フロー] ブランチを展開して、[GetFlights] をクリックします。



- c 右の表示枠で、[入力/チェックポイント] ボタンを選択します。
- d [Output] セクションで、Flight (配列) ノード行の [追加] ボタンをクリックして、Flight[1] 配列を作成します。配列を展開して [FlightNumber] を選択し、[OK] をクリックします。



6 カスタム・コード・ステップのプロパティを作成します。

キャンバスで [カスタム コード] アクティビティを選択し、[プロパティ] 表示枠で [入力/チェックポイント] ビューを開きます。[プロパティの追加] ツールバー・ボタンを展開し、[入力プロパティの追加] を選択します。String タイプのプロパティを FlightInfo という名前で新規に作成します。

7 カスタム・コード・ステップの値を定義します。

ここでは、複数のソースを使用して値を定義します。この例では、CustomerName、定数文字列、OrderNumber を組み合わせた値を設定します。



- a [FlightInfo] 行の右隅にある [データ ソースにリンク] ボタンをクリックします。
[リンク元の選択] ダイアログ・ボックスが開きます。
- b [カスタム式] をクリックして式領域を表示します。
- c [データ ソース カラム] オプションを選択します。ツリーの [WS_Flights!Input] ノードで、[CustomerName] を選択します。[追加] をクリックします。
- d [式] 領域で、既存の式の後に「_OrderNumber_」(下線あり) と入力します。
- e [利用可能なステップ] を選択し、[テスト フロー] ブランチを展開します。
[CreateFlightOrder] ノードを選択します。右の表示枠で、[入力/チェックポイント] ボタンを選択します。下部の表示枠で、[出力] の [Body] ノードを展開し、ツリーを展開して [OrderNumber] 要素を選択し、[追加] をクリックします。



カスタム・コード入力プロパティの **FlightInfo** の値は次のようになります。

```
{DataSource.WS_Flights!Input.CustomerName}_OrderNumber_{Step.OutputProperties.  
StServiceCallActivity(x).Body.CreateFlightOrderResponse.CreateFlightOrderResult.  
OrderNumber}
```

[OK] をクリックして、ダイアログ・ボックスを閉じます。

8 イベントを作成します。

このステップでは、Application Program Interface (API) を使用するために、イベント・ハンドラを作成します。このセクションには C# コードを追加できます。イベントを定義することで、テストをカスタム要件に適合させ、UFT に組み込まれていないアクションを実行できるようになります。この例では、カスタム文字列をレポートに送信するコードを追加します。



キャンバスで**カスタム・コード・ステップ**を選択します。[プロパティ] 表示枠で、[イベント] ボタンをクリックします。[ExecuteEvent] 行のドロップダウン矢印をクリックし、[標準設定のハンドラの作成] を選択します。**CodeActivity(x)_OnExecuteEvent** というイベントが作成され、新しいタブの [TestUserCode.cs] が開きます。

9 アクティビティに割り当てられたインデックスを使用して、「Todo」セクションを編集します。

Todo セクションでコメントになっているテキストを次のように変更します。

```
CodeActivity(x).Report("Customer and Order  
Number",CodeActivity(x).Input.FlightInfo);
```

次の例では、イベントに割り当てられたインデックスは 12 なので、文字列は `CodeActivity12.Report("Customer and Order Number",CodeActivity12.Input.FlightInfo)` になります。

```

13
14 [Serializable()]
15 public class TestUserCode : TestEntities
16 {
17
18     /// <summary>
19     /// CodeActivity12 アクティビティの ExecuteEvent イベントのハンドラ。
20     /// </summary>
21     /// <param name="sender">ExecuteEvent イベントが発生したアクティビティ オブジェクト。</param>
22     /// <param name="args">イベントの引数がアクティビティに渡されました。</param>
23     /// </summary>
24     /// この CodeActivity12 を使用して、CodeActivity12 アクティビティのコンテキスト (入出力プロパティ)
25     public void CodeActivity12_OnExecuteEvent(object sender, SActivityBaseEventArgs args)
26     {
27         CodeActivity12.Report("Customer and Order Number",CodeActivity12.Input.FlightInfo);
28     }
29
30 }
31

```

1 テストを実行し、結果を確認します。

結果をドリルダウンして、[カスタム コード] ステップを表示します。[キャプチャ データ] 表示枠に新しいエントリの [顧客名と注文番号] があることに着目してください。

ヒント: [その他] フォルダの [レポート メッセージ] アクティビティを使用して、テキストとプロパティの値をレポートに送信することもできます。

これからの作業

これで Web サービスのテストを作成できるようになりました。このテストは、ほかのタイプのサービスやアプリケーション・コンポーネントに関連付けることができます。次のレッスンでは、REST サービスのテストを作成するプロセスを順を追って説明します。

レッスン 4

REST サービスのテスト

UFT を使用して、REST サービスをモデル化し、そのテストを作成できます。

このレッスンの内容は次のとおりです。

- ▶ REST サービス・アクティビティを作成する方法
- ▶ REST テストを実行する方法
- ▶ REST メソッドにデータを割り当てる方法
- ▶ 出力を確認する方法
- ▶ REST サービスの変更を解決する方法

REST サービス・アクティビティを作成する方法

ここでは、REST サービス・アクティビティをモデル化する方法を、サンプル・アプリケーションを使用して説明します。REST サービスを使用する際のベスト・プラクティスは、REST サービスの再利用可能なプロトタイプ・メソッドを作成することです。このようなメソッドを作成すると、さまざまなテスト・ステップや別のテストでメソッドを再利用できます。

1 サンプル・フライト・アプリケーションを開始します。

フライト・アプリケーション・サービスが実行中であることを、13 ページ「アプリケーションを起動する方法」の説明に従って確認します。

2 REST サービスのモデリング・ドキュメントを取得します。

サンプル・アプリケーションのコマンド・ウィンドウで、「h」と入力して、ENTER キーを押します。ブラウザが開いて、REST サービスのモデリング情報が表示されます。このファイル (`index.htm`) は、`<インストール・フォルダ>\SampleApplication\Help` フォルダにあります。

3 要求本文をファイルに保存します。

[FlightOrders] > [ReserveOrder (POST)] の要求本文をクリップボードにコピーします。XML コードのみをコピーしてください。

```
<FlightOrderDetails xmlns="HP.SOAQ.SampleApp">
  <Class>Business</Class>
  <CustomerName>John Doe</CustomerName>
  <DepartureDate>2012-12-12</DepartureDate>
  <FlightNumber>1304</FlightNumber>
  <NumberOfTickets>21</NumberOfTickets>
</FlightOrderDetails>
```

テキスト・エディタで新しいファイルを作成し、クリップボードの内容を貼り付けます。ファイルは、`body.xml` として任意の場所に保存します。

4 REST サービスのエンドポイント URL を取得します。

ブラウザ・ウィンドウに戻って、[FlightOrders] > [ReserveOrder (POST)] 操作の URL (`http://localhost:24240/HPFlights_REST/FlightOrders/`) をクリップボードにコピーします。

5 新しいテストを作成します。

- a [ファイル] > [新規] > [テスト] を選択し, [API テスト] タイプを選択します。
- b **RESTServiceTest** を新しいテストの名前に指定します。
- c [作成] をクリックします。

6 REST サービスを追加します。



- a [REST サービスの追加] ツールバー・ボタンをクリックします。[REST サービスの追加] ダイアログ・ボックスが開きます。
- b 左の表示枠で[新規サービス]ノードをクリックし, その名前を **SampleRESTService** に変更します。

7 リソースを追加します。

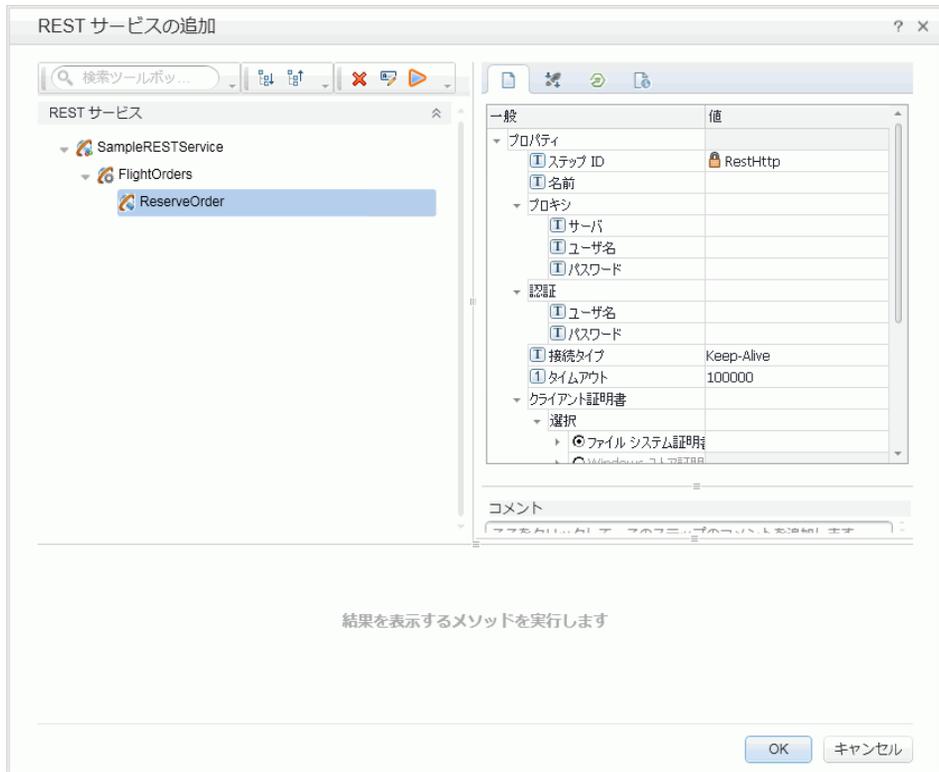


ツールバーの [リソースの追加] ボタンをクリックし, リソースの名前を **FlightOrders** に変更します。

8 メソッドを REST サービスに追加します。



[**メソッドの追加**] ボタンをクリックし、メソッドの名前を **ReserveOrder** に変更します。



9 メソッドの HTTP プロパティを設定します。



右の表示枠で [**HTTP 入力/チェックポイント**] ビューを開きます。

a クリップボードの内容を [**URL**] 行に貼り付けます。

b [**HTTP メソッド**] を **POST** に設定します。



c [**Request Headers (配列)**] 行の [**追加**] ボタンをクリックして、配列要素を追加します。

d [Request Headers (配列)] 配列を展開します。ヘルプ・ページのメソッドの詳細を使用して、ヘッダの名前と値を次のように設定します。

- [名前] 行 : Content-Type
- [値] 行 : text/xml

入力	値
▼ プロパティ	
URL	http://localhost:24240/HPFlights
HTTP メソッド	POST
HTTP バージョン	1.1
▼ RequestHeaders (配列) +	
▼ RequestHeaders[1] ✖	
名前	Content-Type
値	text/xml

10 入力プロパティを作成します。



a 右の表示枠で [入力/チェックポイント] ビューを開きます。



b [追加] > [入力プロパティの追加] を選択します。

c **Class** という **String** タイプのプロパティを追加します。

d さらに、**Customer_Name** という **String** タイプのプロパティを追加します。

e さらに、**Departure_Date** という **Date Time** タイプのプロパティを作成します。

f さらに、**Flight_Number** という **Int** タイプのプロパティを追加します。

g さらに、**Number_of_Tickets** という **Int** タイプのプロパティを追加します。

11 出力プロパティを作成します。



a [追加] > [出力プロパティの追加] を選択します。

b さらに、**Total_Price** という **Int** タイプのプロパティを追加します。

c さらに、**Order_Number** という **Int** タイプのプロパティを追加します。

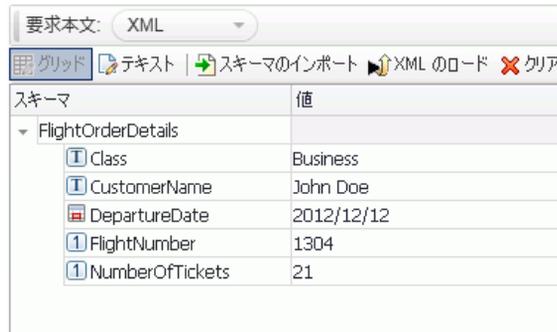
12 要求本文をインポートします。



a 右の表示枠で [HTTP] ビューを開きます。

b [要求本文] タイプとして [XML] を選択します。

- c [XML のロード] ボタンをクリックし、前に保存した **body.xml** ファイルをロードします。



13 メソッドをテストする。



[メソッドの実行] ツールバー・ボタンをクリックし、メソッドの有効性を確認します。結果をスクロールし、応答本文に注文番号と価格が含まれていることを確認します。

HTTPバージョン	1.1
通信時間	0:0.418
応答本文	<pre> 1: <CreatedOrderParams xmlns="HP.SOAQ.SampleApp" xmlns:i="http://www.w3.org/2001/XMLSchema-instance"> 2: <OrderNumber>20</OrderNumber> 3: <TotalPrice>3717</TotalPrice> 4: </CreatedOrderParams> </pre>

14 メソッドをツールボックスに追加します。

[REST サービスの追加] ダイアログ・ボックスで [OK] をクリックします。REST デザイナによって、REST サービスおよびそのリソースとメソッドが [ツールボックス] 表示枠の [ローカル アクティビティ] カテゴリに追加されます。

15 すべてのテストでアクティビティを使用できるように、REST アクティビティを共有します。

[**ツールボックス**] 表示枠で、REST サービスの親ノード **SampleRESTService** を選択し、右クリック・メニューから [**移動先**] > [**ファイル システムのアクティビティ**] を選択します。これで、REST サービス・アクティビティが [**ツールボックス**] 表示枠の [**ファイル システムのアクティビティ**] セクションに移動しました。



入力パラメータと HTTP 情報が含まれている、REST サービスのプロトタイプ・アクティビティが作成されました。これで、アクティビティをテストにドラッグして、ほとんど介在せずに実行できます。

REST テストを実行する方法

ここでは、前の項で作成した REST サービス・メソッドを実行する方法を説明します。製品に付属するサンプル・データ・ファイルを使用して、テストにデータを組み込みます。

1 テスト・ステップを作成します。

[ファイル システムのアクティビティ] > [REST サービス] > [SampleRESTService] > [FlightOrders] > [ReserveOrder] メソッドをダブルクリックして、キャンバスのテスト・フローに追加します。

2 REST メソッド・ラッパーの内部の HTTP 要求を展開します。

[ReserveOrder] メソッド・フレームの左側の矢印をクリックし、内側の HTTP 要求フレームを表示します。HTTP 要求フレーム内をクリックし、[プロパティ] 表示枠に [HTTP] ビューを開きます。



このプロパティ値は、前に **body.xml** ファイルからインポートした値です。この値は、実行したばかりのテストで使用された値です。

The screenshot shows the 'テストフロー' (Test Flow) editor. A 'ReserveOrder8' step is highlighted, and its internal 'HTTP 要求9' (HTTP Request 9) is expanded. The '要求本文: XML' (Request Body: XML) field is visible. Below it, a table shows the 'スキーマ' (Schema) details for 'FlightOrderDetails'.

スキーマ	値
FlightOrderDetails	
Class	Business
CustomerName	John Doe
DepartureDate	2012/12/26
FlightNumber	1304
NumberOfTickets	21

3 テストを実行します。



[実行] > [実行] を選択して、テストを実行します。

4 レポートを表示します。



Run Result Viewer の左枠で、[すべて展開] ボタンをクリックするか、右クリック・メニューから [すべて展開] を選択し、[HTTP 要求] ノードを選択します。

[キャプチャ データ] 表示枠の [ResponseBody] リンクをクリックして、別のブラウザ・ウィンドウで応答を開きます。

キャプチャ データ	
RequestHeader_Host	localhost:24240
RequestHeader_Content-Length	250
RequestHeader_Expect	100-continue
RequestHeader_Connection	Keep-Alive
ResponseHeader_Content-Length	177
ResponseHeader_Content-Type	application/xml; charset=utf-8
ResponseHeader_Date	Fri, 21 Dec 2012 01:53:32 GMT
ResponseHeader_Server	Microsoft-HTTPAPI/2.0
ResponseBody	<CreatedOrderParams ... /CreatedOrderParams>
メッセージ	HTTP 要求を正常に呼び出しました
HTTP メソッド	'POST'

[応答本文] 行に、**OrderNumber** および **TotalPrice** の要素が含まれていることを確認します。これは、次の REST サービス・ヘルプ・ページの操作説明に対応しています。航空券の申し込みを新規に作成し、OrderNumber および TotalPrice を返します。

```

_ <CreatedOrderParams xmlns="HP.SOAQ.SampleApp"
  xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
  <OrderNumber>18</OrderNumber>
  <TotalPrice>3717</TotalPrice>
</CreatedOrderParams>
    
```

REST メソッドにデータを割り当てる方法

データをメソッドに割り当てるために、内部 HTTP 要求プロパティを外部 REST ラッパー・プロパティにリンクします。これにより、上位レベルである REST メソッドからの要求メッセージのプロパティを制御できます。

REST メソッド・ステップにデータを割り当てる

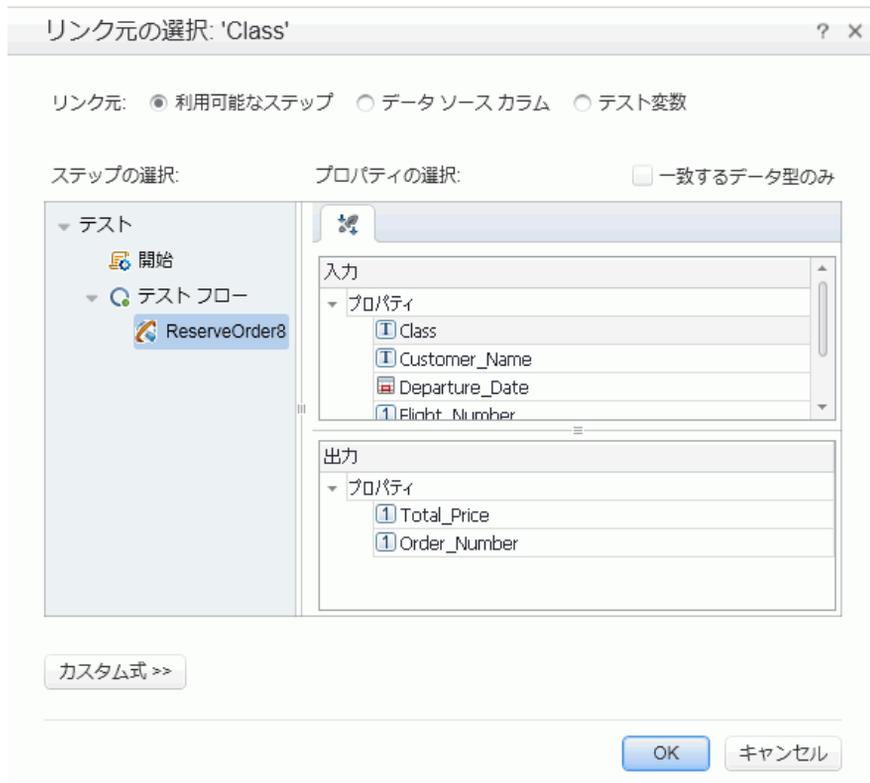


1 内部 [HTTP 要求] フレームを選択し、[プロパティ] 表示枠に [HTTP] ビューを開きます。



2 [Class] 行の右隅にある [データ ソースにリンク] ボタンをクリックします。[リンク元の選択] ダイアログ・ボックスが開きます。

3 [利用可能なステップ] をデータ・ソースとして選択し、左の表示枠で [ReserveOrder] ノードを選択し、右の表示枠で [Class] プロパティを選択します。[OK] をクリックします。



- 4 これを、上記ステップのほかの入力プロパティ (**CustomerName**, **DepartureDate**, **FlightNumber**, **NumberOfTickets**) についても繰り返します。その結果、[HTTP] ビューに新しいリンクが表示されます。

スキーマ	値
<ul style="list-style-type: none"> ▼ FlightOrderDetails <ul style="list-style-type: none"> Class {Step.InputProperties.RESTActivity8.Class} CustomerName {Step.InputProperties.RESTActivity8.Customer_Name} DepartureDate {Step.InputProperties.RESTActivity8.Departure_Date} FlightNumber {Step.InputProperties.RESTActivity8.Flight_Number} NumberOfTickets {Step.InputProperties.RESTActivity8.Number_of_Tickets} 	

5 サンプル・データをインポートします。

REST メソッドのプロパティに使用する値を持ったデータ・ソースをインポートします。



- a ウィンドウ下部の [データ] 表示枠で、[データ ソース] ボタンを展開し、[Excel] を選択します。[新規 Excel データ ソース] ダイアログ・ボックスが開きます。



- b [Excel ファイルのパス] フィールドの横の [参照] ボタンをクリックし、<インストール・ディレクトリ>\SampleApplication フォルダにあるサンプル・アプリケーションの SampAppData Excel ファイルを指定します。

- c サンプルにはヘッダ行が含まれているため、[Excel ファイルにヘッダ行が含まれています] チェック・ボックスを選択します。

- d [データ ソース名] フィールドに「REST_Flights」と入力します。

- e [Excel ファイルのコピーを作成] オプションを選択します。これにより、データ・ファイルのコピーがテストとともに保存されます。

- f [OK] をクリックします。データが [データ] 表示枠にインポートされます。

6 REST メソッドのプロパティをデータ・ソースにリンクします。

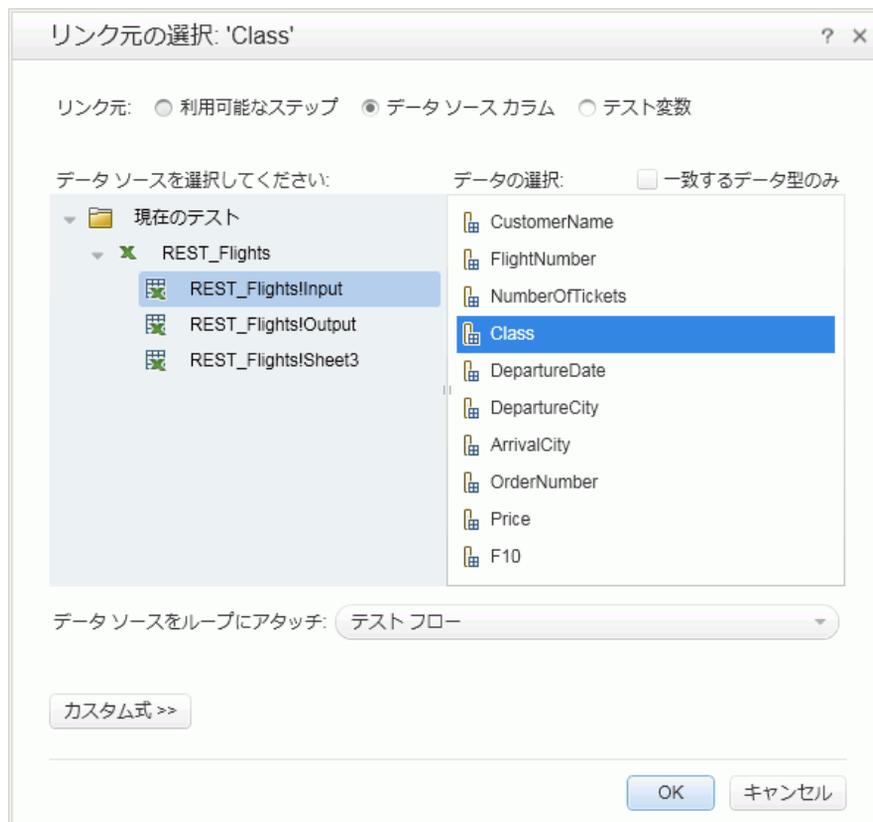


- a 外部 [ReserveOrder] フレームを選択し、右の表示枠に [入力/チェックポイント] ビューを開きます。



- b [Class] 行の最も右隅にある [データ ソースにリンク] ボタンをクリックします。[リンク元の選択] ダイアログ・ボックスが開きます。

- c [データ ソース カラム]をデータ・ソースとして選択し、左の表示枠で[ReserveOrder]ノードを選択し、右の表示枠で [Class] プロパティを選択します。[OK] をクリックします。



- d ほかの入力プロパティについて、上記の手順を繰り返します。(Customer_Name, Departure_Date, Flight_Number)
- e Number_of_Tickets に、「2」と入力します。

f その結果, [プロパティ] ビューに新しいリンクが表示されます。

入力	値
プロパティ	
Class	{DataSource.REST_FlightsInput.Class}
Customer_Name	{DataSource.REST_FlightsInput.CustomerNa}
Departure_Date	{DataSource.REST_FlightsInput.DepartureDat}
Flight_Number	{DataSource.REST_FlightsInput.FlightNumber}
Number_of_Tickets	2

7 反復回数を設定します。

反復回数を設定をすることで, REST メソッドが複数のデータ・セットをデータ・ソースから使用する方法についてわかります。



a キャンバスで [テスト フロー] フレームを選択し, 右の表示枠で [入力/チェックポイント] ビューを開きます。

b 'For' ループを反復 3 回で設定します。

'For' ループ
 'Do While' ループ
 'For Each' ループ

反復回数:

8 テストを実行します。



[実行] > [実行] を選択して, テストを実行します。

9 要求でテーブル・データが使用されたことを確認します。



テストを実行します。Run Result Viewer の左枠で、**[すべて展開]** ボタンをクリックし、**[HTTP 要求]** ノードを選択します。[キャプチャ データ] 表示枠の **[要求本文]** リンクをクリックします。割り当てられたプロパティに対して、[データ] 表示枠のデータがテストで使用されたことを確認します。(Class, CustomerName, DepartureDate, FlightNumber)

```
<FlightOrderDetails xmlns="HP.SOAQ.SampleApp">
  <Class>Business</Class>
  <CustomerName>John Freeman</CustomerName>
  <DepartureDate>2012-12-12</DepartureDate>
  <FlightNumber>1304</FlightNumber>
  <NumberOfTickets>21</NumberOfTickets>
</FlightOrderDetails>
```

10 応答データを保存します。

この実行の応答データを保存し、以降の手順で使用します。

- a** Run Result Viewer の左枠で、**[HTTP 要求]** ノードを選択します。[キャプチャ データ] 表示枠の **[応答本文]** リンクをクリックします。ブラウザ・ウィンドウが開いて、XML 応答が表示されます。
- b** ウィンドウの全体の内容をファイル **Response.xml** に保存します。Run Results Viewer を閉じます。

出力を確認する方法

REST メソッドの出力が正しいことを確認するために、チェックポイントを設定します。

1 チェックポイントを挿入します。



- a 内側の **[HTTP 要求]** フレームをクリックします。[プロパティ] 表示枠で、**[HTTP]** ビューを開きます。
- b 表示枠の下のセクションには、出力プロパティまたは出力スキーマが表示されます。これらは、サーバーの応答を確認するチェックポイントとして使用します。下の表示枠で、**[要求本文]** ドロップダウンから **[XML]** を選択します。
- c **[XML のロード]** ボタンをクリックし、前に保存した **Response.xml** ファイルをロードします。**[OrderNumber]** および **[TotalPrice]** の行の **[検証]** チェック・ボックスを選択します。
- d **[OrderNumber]** の値を 10 より大きい数 (> 10) に設定し、**[TotalPrice]** の値を 255 未満 (< 255) に設定します。

スキーマ	検証	値
CreatedOrderParams	<input type="checkbox"/>	=
① OrderNumber	<input checked="" type="checkbox"/>	> 10
① TotalPrice	<input checked="" type="checkbox"/>	< 250

2 テストを実行します。



[実行] > **[実行]** を選択して、テストを実行します。

3 チェックポイントがパスしたことを確認します。



テストを実行します。Run Result Viewer の左枠で、[すべて展開] ボタンをクリックします。パスしたチェックポイントと、そうでないチェックポイントがあります。



4 チェックポイントが失敗した原因を特定します。

失敗したチェックポイント・ノードのいずれかを選択します。右の表示枠で、[キャプチャ・データ] 表示枠を表示し、実際の結果と期待値を確認します。次の例では、**OrderNumber**は有効ですが、**TotalPrice**は255を超えているため、無効です。

名前	結果	プロパティ	実際の結果	評価スタイル	期待値	詳細
"チェックポイント 1"	✓	"CreatedOrderParams[1]/TotalPrice[1]"	"48"	>	"10"	""
"チェックポイント 2"	✗	"CreatedOrderParams[1]/OrderNumber[1]"	"392"	<	"250"	""

5 テストを保存します。

Run Results Viewer を閉じて、テストを保存します。

REST サービスの変更を解決する方法

最初に、URL やプロパティ名など特定のプロパティで、REST サービス・メソッド [ReserveOrder] のプロトタイプを作成しました。テストを作成した後にこれらのプロパティを変更すると、テスト・ステップがプロトタイプと一致しくなくなります。競合の解決ウィザードは、メソッドのプロパティへの変更を検出し、解決に役立てることができます。

- 1 [ツールボックス] 表示枠で [ReserveOrder] ノードを選択し、右クリック・メニューから [サービスの編集] を選択します。[REST サービスの編集] ダイアログ・ボックスが開きます。



- 2 [ReserveOrder] メソッドを選択し、[プロパティ] 表示枠で [カスタム入力/チェックポイント] ビューを開きます。



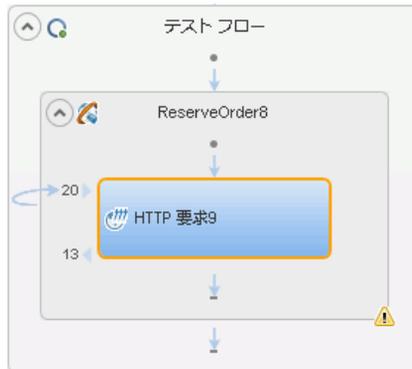
- 3 [Flight_Number] プロパティを選択し、[プロパティの編集] ツールバー・ボタンをクリックします。[プロパティの編集] ダイアログ・ボックスから、プロパティ名を「Flight_Number_1」に変更します。[OK] をクリックします。



- 4 [Number_of_Tickets] プロパティを選択し、[プロパティの削除] ツールバー・ボタンをクリックします。警告を確認し、[OK] をクリックします。

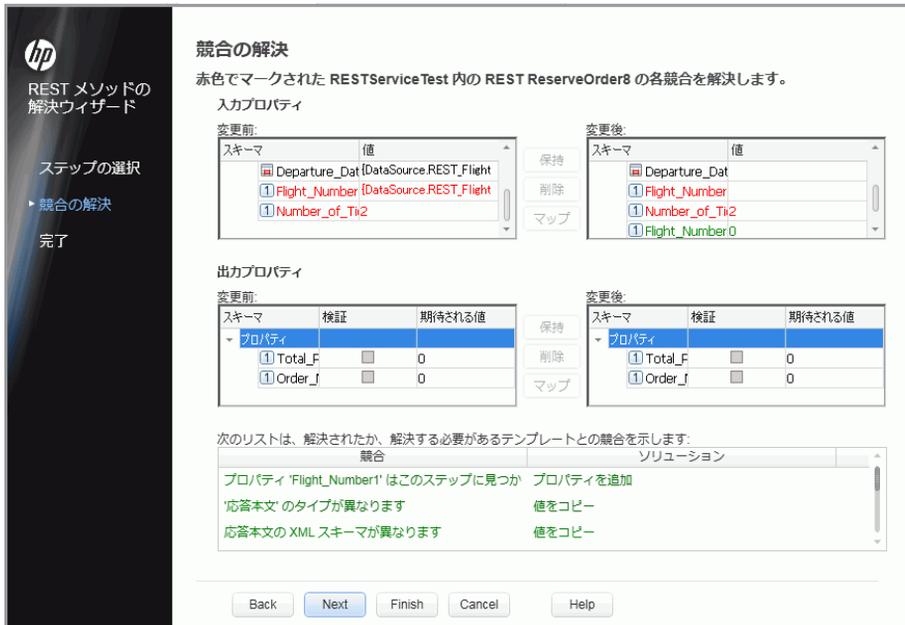
- 5 [REST サービスの編集] ダイアログ・ボックスで [OK] をクリックします。

- 6 キャンパスの表示 REST メソッド・フレームの右下隅にアラート・アイコンが表示されます。



- 7 アラート・アイコンの横にあるドロップダウン矢印をクリックし、テキスト・メッセージを選択します。(このステップは解決する必要があります。ステップの解決…) REST メソッドの解決ウィザードが開きます。

- 8 このウィザードの最初の画面に、問題のあるステップが表示されます。複数のステップが影響を受けている場合、解決するステップと無視するステップを選択できます。[次へ]をクリックします。
- 9 [競合の解決] 画面で、左の表示枠から [Number_of_Tickets] プロパティを選択します。[維持] をクリックします。これにより、メソッドのプロトタイプから削除された場合でも、既存のステップはプロパティを維持します。



- 10 [変更後] 表示枠で新規プロパティ [Flight_Number_1] プロパティを選択し、**削除** をクリックします。

ウィザード画面の下のセクションをスクロールし、すべての競合およびその解決方法のログを表示します。

- 11 [次へ] をクリックします。[完了] をクリックして、ウィザードを終了し、テストに戻ります。

これからの作業

標準的なアクティビティ、Web サービス、REST サービスのテストを作成できるようになったので、GUI のないアプリケーション用に独自のテストを作成できます。