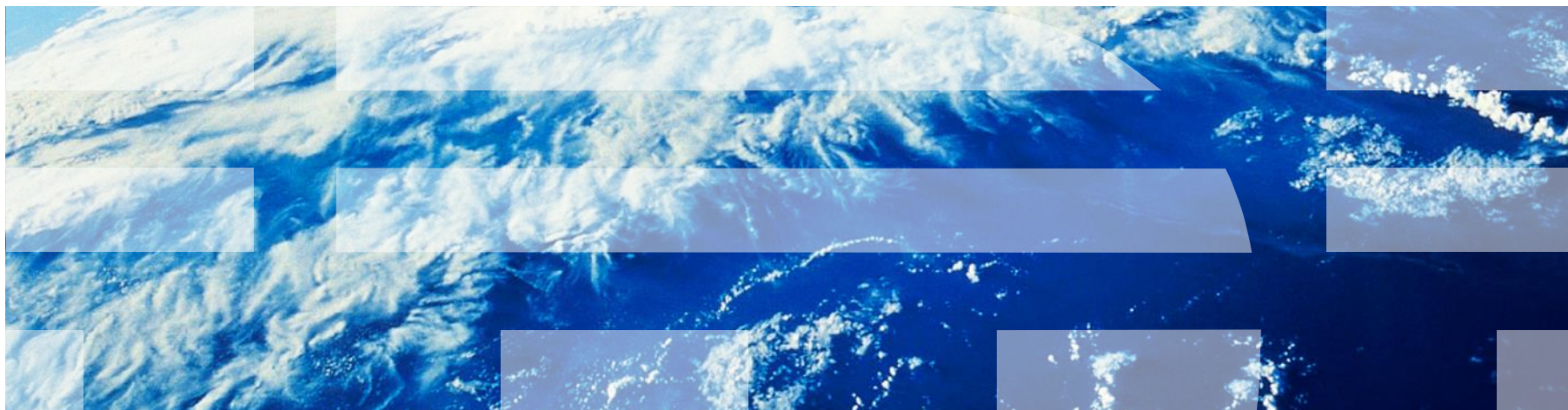


# ***IBM Worklight V6.1.0*** **入門**

## HTTP アダプター – HTTP バックエンド・システムとの通信



## 商標

- IBM、IBM ロゴおよび [ibm.com](http://www.ibm.com) は、世界の多くの国で登録された International Business Machines Corporation の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、<http://www.ibm.com/legal/copytrade.shtml> をご覧ください。
- Java およびすべての Java 関連の商標およびロゴは Oracle やその関連会社の米国およびその他の国における商標または登録商標です。
- この資料は、事前に IBM の書面による許可を得ずにその一部または全部を複製することは禁じられています。

## IBM® について

- <http://www.ibm.com/ibm/us/en/> を参照してください。

# アジェンダ

- HTTP アダプターとは
- HTTP アダプターを使用するための準備
- アダプターの作成
- SOAP の使用
  - SOAP ベースのサービス要求の作成
  - サービス要求の呼び出し
  - サービスのディスカバリー
- バックエンド・サービスのディスカバリー
- 演習

## HTTP アダプターとは

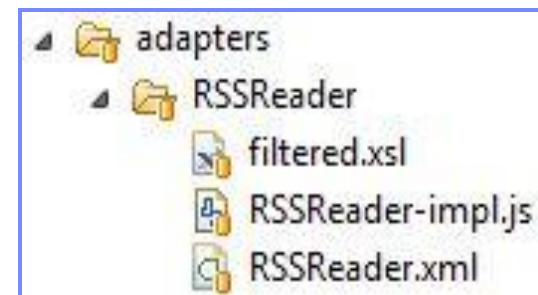
- Worklight® HTTP アダプターには以下の特長があります。
  - RESTful サービスおよび SOAP ベースのサービスと連携します。
  - 構造化 HTTP ソース (例えば、RSS フィード) を読み取ることができます。
  - GET 要求や POST HTTP 要求の送信を可能にし、応答ヘッダーおよび本体からデータを取得します。
  - 単純なサーバー・サイド JavaScript™ を使用して簡単にカスタマイズできます。
  - オプションでサーバー・サイド・フィルタリングを使用することができます。
  - データは、XML、HTML、JSON、プレーン・テキストのいずれかの形式で取得することができます。

# アジェンダ

- HTTP アダプターとは
- HTTP アダプターを使用するための準備
- アダプターの作成
- SOAP の使用
  - SOAP ベースのサービス要求の作成
  - サービス要求の呼び出し
- バックエンド・サービスのディスカバリー
- 演習

## HTTP アダプターを使用するための準備

- XML を使用してアダプターを構成します。
- XML を使用してアダプターのプロパティとプロシージャーを定義します。
- JavaScript を使用してプロシージャーを作成します。
- オプション: アダプターで受け取ったレコードとフィールドを XSL を使用してフィルターに掛けます。

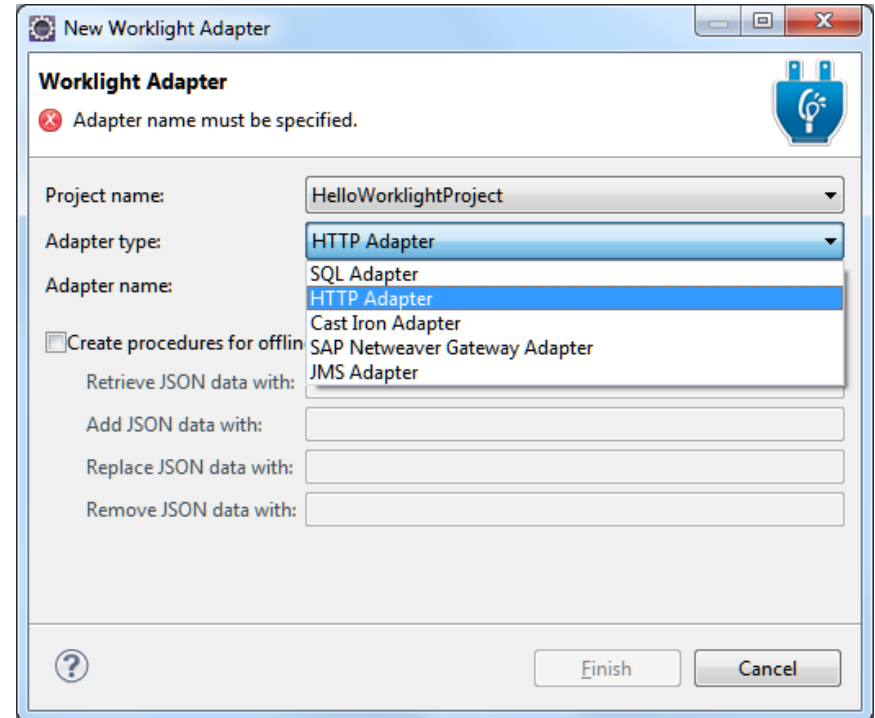
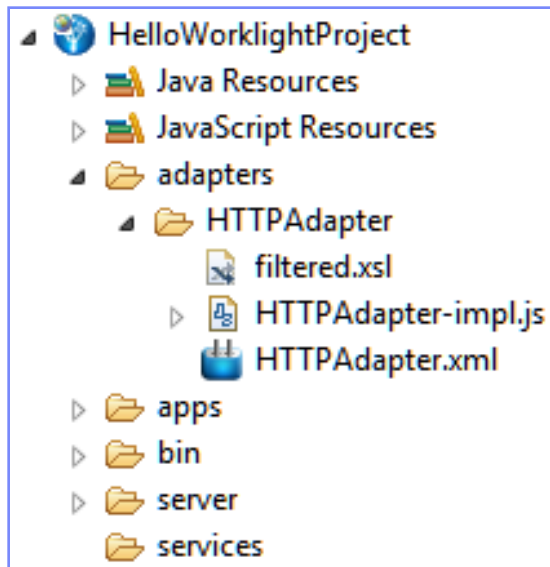


# アジェンダ

- HTTP アダプターとは
- HTTP アダプターを使用するための準備
- **アダプターの作成**
- SOAP の使用
  - SOAP ベースのサービス要求の作成
  - サービス要求の呼び出し
- バックエンド・サービスのディスカバリー
- 演習

# アダプターの作成

- Worklight Studio で、Worklight アダプターを作成します。
  - アダプター・タイプとして「HTTP アダプター (HTTP Adapter)」を選択します。
  - 以下のように、標準の HTTP アダプター構造が作成されます。

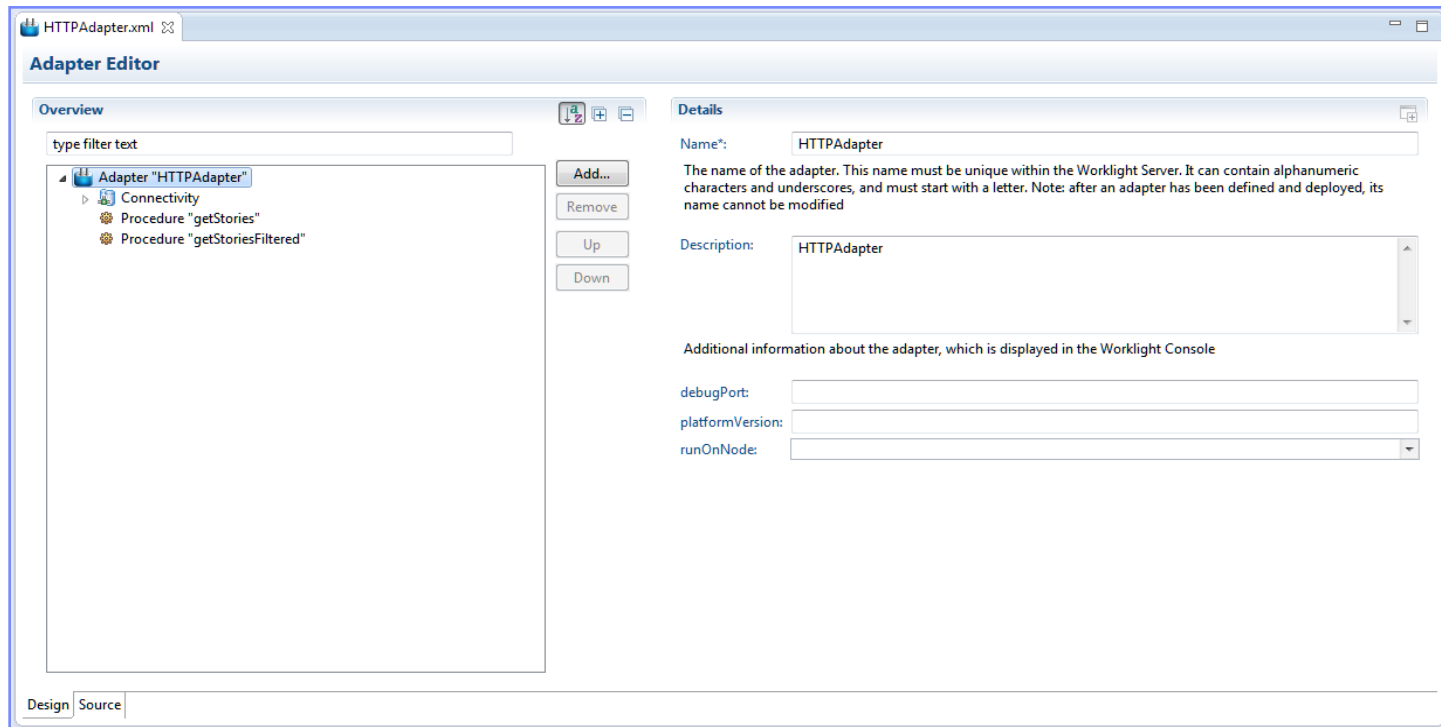




# アダプターの作成 – 続き

## アダプター XML エディター

- 設定およびメタデータはアダプター XML ファイルに保管されます。
- アダプター XML ファイルは、設計エディターとソース・エディターのいずれかを使用して変更することができます。



## アダプターの作成 – 続き

### XML ファイル: 接続設定

- アダプター XML ファイルを編集するには、以下のようにしてください。
  - プロトコルを HTTP または HTTPS に設定します。
  - HTTP ドメインを HTTP URL のドメイン部に設定します。
  - TCP ポートを設定します。

```
<connectivity>
  <connectionPolicy xsi:type="http:HTTPConnectionPolicyType">
    <protocol>http</protocol>
    <domain>rss.cnn.com</domain>
    <port>80</port>
    <!-- Following properties used by adapter's key manager
    <sslCertificateAlias></sslCertificateAlias>
    <sslCertificatePassword></sslCertificatePassword>
    -->
  </connectionPolicy>
  <loadConstraints maxConcurrentConnectionsPerNode="2" />
</connectivity>
```

## アダプターの作成 – 続き

### XML ファイル: プロシーチャーの宣言

- アダプター XML ファイルを編集します。
- 必要なプロシーチャーを宣言します。

```
<wl:adapter name="RSSReader"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:wl="http://www.worklight.com/integration"
  xmlns:http="http://www.worklight.com/integration/http">

  <displayName>RSSReader</displayName>
  <description>RSSReader</description>
  <connectivity>
    <connectionPolicy xsi:type="http:HTTPConnectionPolicyType">
      <protocol>http</protocol>
      <domain>rss.cnn.com</domain>
      <port>80</port>
      <!-- Following properties used by adapter's key manager
      <sslCertificateAlias></sslCertificateAlias>
      <sslCertificatePassword></sslCertificatePassword>
      -->
    </connectionPolicy>
    <loadConstraints maxConcurrentConnectionsPerNode="2" />
  </connectivity>

  <procedure name="getStories"/>
  <procedure name="getStoriesFiltered"/>

</wl:adapter>
```

## アダプターの作成 – 続き

### JS ファイル: プロシージャーの実装

- プロシージャーはアダプター JavaScript ファイルで実装されます。
- サービス URL を使用してプロシージャーが呼び出されます。
- URL の一部は、例えば、<http://example.com/> のように、固定のものになります。それらは XML ファイルで宣言されます。
- URL のその他の部分はパラメーター化することができます。つまり、Worklight プロシージャーに指定されたパラメーター値によって実行時に置換されるということです。
- パラメーター化できる URL 部は、以下のとおりです。
  - パス・エレメント
  - 照会ストリング・パラメーター
  - フラグメント

http://example.com/rest/customers?custid=12

照会ストリング

パス・エレメント

- アダプターの拡張オプション (cookie、header、encoding など) については、IBM Worklight ユーザー文書を参照してください。

## アダプターの作成 – 続き

### JS ファイル: プロシージャーの実装

- JavaScript ファイルで指定するプロシージャー名は XML ファイルで指定するものと同じでなければなりません。
- 必須指定の呼び出しパラメーターは、method、path、および returnedContentType です。
- プロシージャーは実行時にパラメーター化することができます。

```
<connectivity>
  <connectionPolicy xsi:type="http:HT
    <protocol>http</protocol>
    <domain>rss.cnn.com</domain>
    <port>80</port>
  </connectionPolicy>
  <loadConstraints maxConcurrentConne
</connectivity>
<procedure name="getStories"/>
<procedure name="getStoriesFiltered"/>
```

プロシージャー XML ファイル

```
function getStories(interest) {
  path = getPath(interest);

  var input = {
    method : 'get',
    returnedContentType : 'xml',
    path : path
  };

  return WL.Server.invokeHttp(input);
}
```

プロシージャー JavaScript ファイル

## アダプターの作成 – 続き

### JS ファイル: プロシージャーの実装

- HTTP 要求を呼び出すには、`WL.Server.invokeHttp` メソッドを使用します。
- 入力パラメーター・オブジェクトを指定します。
- 以下のものを指定する必要があります。
  - HTTP メソッド: GET または POST
  - 返されるコンテンツのタイプ:  
XML、JSON、HTML、または  
プレーン・テキスト
  - サービス・パス
  - 照会パラメーター (任意指定)
  - 要求本体 (任意指定)
  - 変換タイプ (任意指定)
- すべての呼び出しオプションを確認するには、IBM Worklight ユーザー文書を参照してください。

```
function getStoriesFiltered(interest) {  
    path = getPath(interest);  
  
    var input = {  
        method : 'get',  
        returnedContentType : 'xml',  
        path : path,  
        transformation : {  
            type : 'xslFile',  
            xslFile : 'filtered.xsl'  
        }  
    };  
  
    return WL.Server.invokeHttp(input);  
}
```

## アダプターの作成 – 続き

### XSL Transformation フィルタリング

- 受け取ったデータに XSL Transformation を適用することができます。
- XSL Transformation は、受け取ったデータのフィルタリングに使用することができます。
- 適用するには、変換オプションをプロシージャラー呼び出し入力パラメーターで指定します。

```
function getStoriesFiltered(interest) {  
  path = getPath(interest);  
  
  var input = {  
    method : 'get',  
    returnedContentType : 'xml',  
    path : path,  
    transformation : {  
      type : 'xslFile',  
      xslFile : 'filtered.xsl'  
    }  
  };  
  
  return WL.Server.invokeHttp(input);  
}
```

# アジェンダ

- HTTP アダプターとは
- HTTP アダプターを使用するための準備
- アダプターの作成
- SOAP の使用
  - SOAP ベースのサービス要求の作成
  - サービス要求の呼び出し
- バックエンド・サービスのディスカバリー
- 演習



## SOAP ベースのサービス要求の作成

- `WL.Server.invokeHttp` メソッドを使用して SOAP エンベロープを作成すると、SOAP エンベロープを直接送信することができます。
  - HTTP アダプターで SOAP ベースのサービスを呼び出すには、SOAP XML エンベロープを要求本体内でエンコードする必要があります。
  - JavaScript 内で XML をエンコードする方法は簡単であり、正式には JavaScript 1.6 の一部である、E4X を使用します。
  - このテクノロジーを使用して、SOAP エンベロープだけでなく、あらゆるタイプの XML ドキュメントをエンコードすることができます。
- SOAP エンベロープに関するエラー・メッセージが発行される場合は、JavaScript バリデーターを無効にしてください。
  - 「プロジェクト (Project)」 > 「プロパティ (Properties)」 > 「ビルダー (Builders)」をクリックして、「JavaScript バリデーター (JavaScript Validator)」チェック・ボックスをクリアします。

## SOAP ベースのサービス要求の作成 – 続き

- JavaScript を使用して SOAP エンベロープを作成します。

```
var request =
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
"http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <CelsiusToFahrenheit xmlns="http://tempuri.org/">
      <Celsius>{celsiusTemp}</Celsius>
    </CelsiusToFahrenheit>
  </soap:Body>
</soap:Envelope>;
```

## SOAP ベースのサービス要求の作成 – 続き

- JavaScript コードおよび変数を SOAP XML に挿入することができます。それらは実行時に評価されます。

```
var request =
<soap:Envelope
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <messageHeader>
      <originatingIP>
        { WL.Server.configuration["local.IPAddress"] }
      </originatingIP>
      <requestTimestamp>
        { new Date().toLocaleString() }
      </requestTimestamp>
    </messageHeader>

    <ConversionRate xmlns="http://www.webserviceX.NET/">
      <FromCurrency>USD</FromCurrency>
      <ToCurrency>EUR</ToCurrency>
    </ConversionRate>
  </soap:Body>
</soap:Envelope>;
```

## サービス要求の呼び出し

- `WL.Server.invokeHttp(options)` メソッドを使用して SOAP サービス要求を呼び出します。
- options オブジェクトには以下のプロパティを含める必要があります。
  - method – 通常は POST。
  - returnedContentType – 通常は XML。
  - path – サービス・パス。
  - body:
    - content – ストリングとしての SOAP XML
    - contentType

```
var options = {
  method : 'post',
  returnedContentType : 'xml',
  path : '/webservices/tempconvert.asmx',
  body: {
    content: request.toString(),
    contentType: 'text/xml; charset=utf-8'
  }
};
```

## サービス要求の呼び出し - 続き

- フル SOAP ベースのサービス呼び出しプロシーチャーの例:

```
function temperatureConvertor(celsiusTemp) {  
    // Use USD and EUR as fromCurrency and toCurrency variables. Case is impo  
  
    var request =  
        <soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
            "http://schemas.xmlsoap.org/soap/envelope/">  
            <soap:Body>  
                <CelsiusToFahrenheit xmlns="http://tempuri.org/">  
                    <Celsius>{celsiusTemp}</Celsius>  
                </CelsiusToFahrenheit>  
            </soap:Body>  
        </soap:Envelope>;  
  
    var input = {  
        method : 'post',  
        returnedContentType : 'xml',  
        path : '/webservices/tempconvert.asmx',  
        body: {  
            content: request.toString(),  
            contentType: 'text/xml; charset=utf-8'  
        }  
    };  
  
    var result = WL.Server.invokeHttp(input);  
  
    return result.Envelope.Body;  
}
```

SOAP エンベロープ

オプション

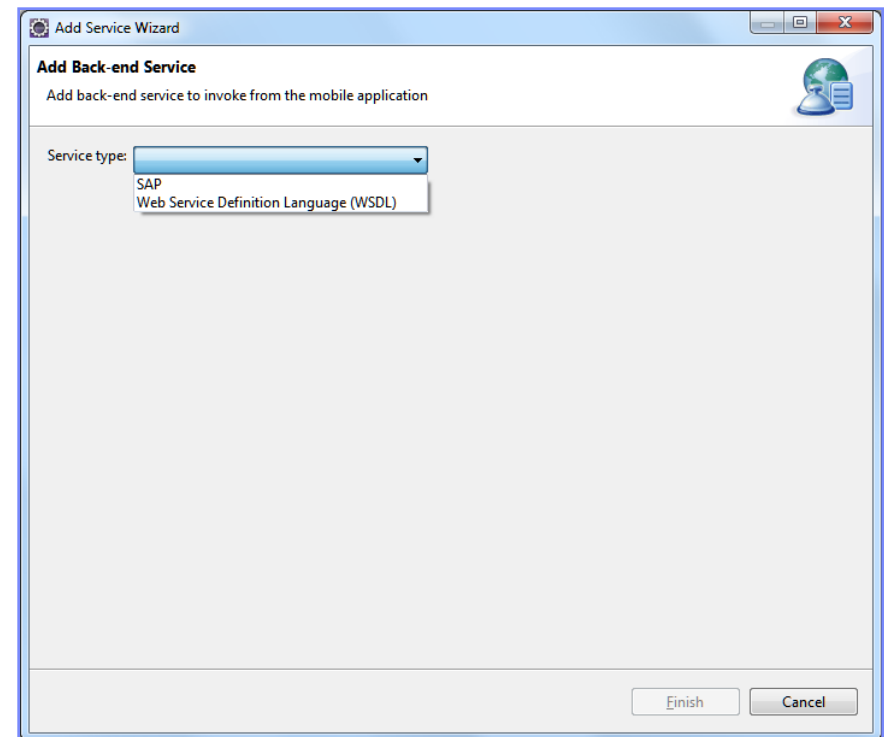
要求の呼び出し

# アジェンダ

- HTTP アダプターとは
- HTTP アダプターを使用するための準備
- アダプターの作成
- SOAP の使用
  - SOAP ベースのサービス要求の作成
  - サービス要求の呼び出し
- バックエンド・サービスのディスカバリー
  - 演習

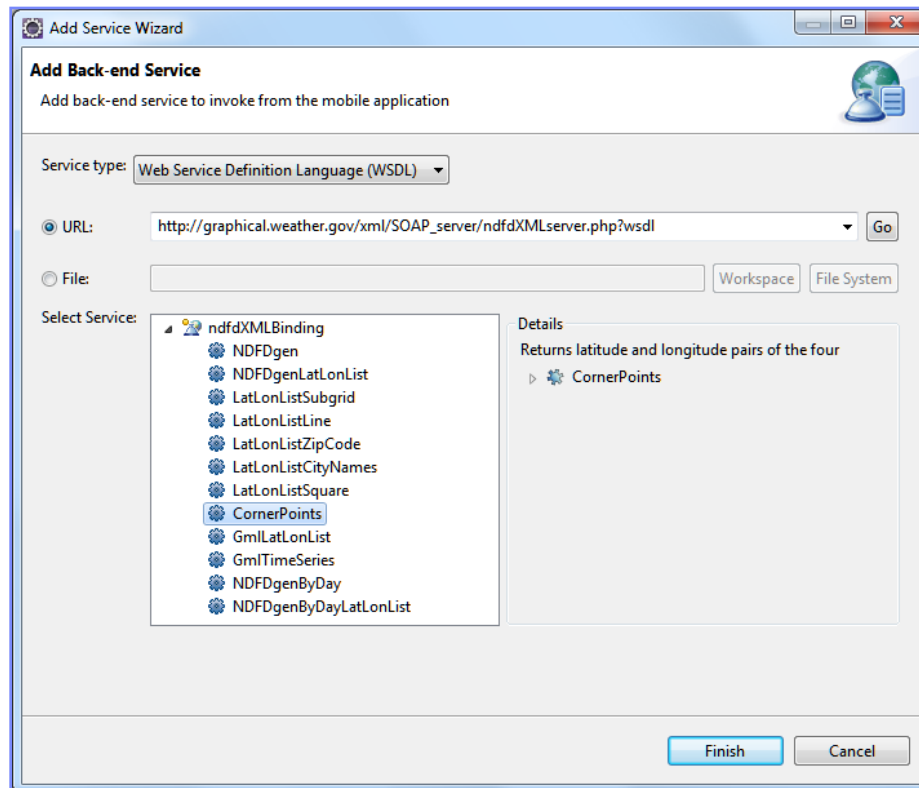
## バックエンド・サービスのディスカバリー

- SOAP (または SAP) サービス用の HTTP アダプターを開発している場合は、「バックエンド・サービスのディスカバリー (Discover Back-end Services)」ツールを使用して、指定されている WSDL に基づいてアダプターをプロシージャーと共に自動生成することにより、開発時間を短縮できます。
- Worklight プロジェクトで「**services**」フォルダーを右クリックして、「バックエンド・サービスのディスカバリー (Discover back-end services)」を選択します。
- サービスのタイプとして「**SAP**」または「**SOAP**」を選択します。



## バックエンド・サービスのディスカバリー – 続き

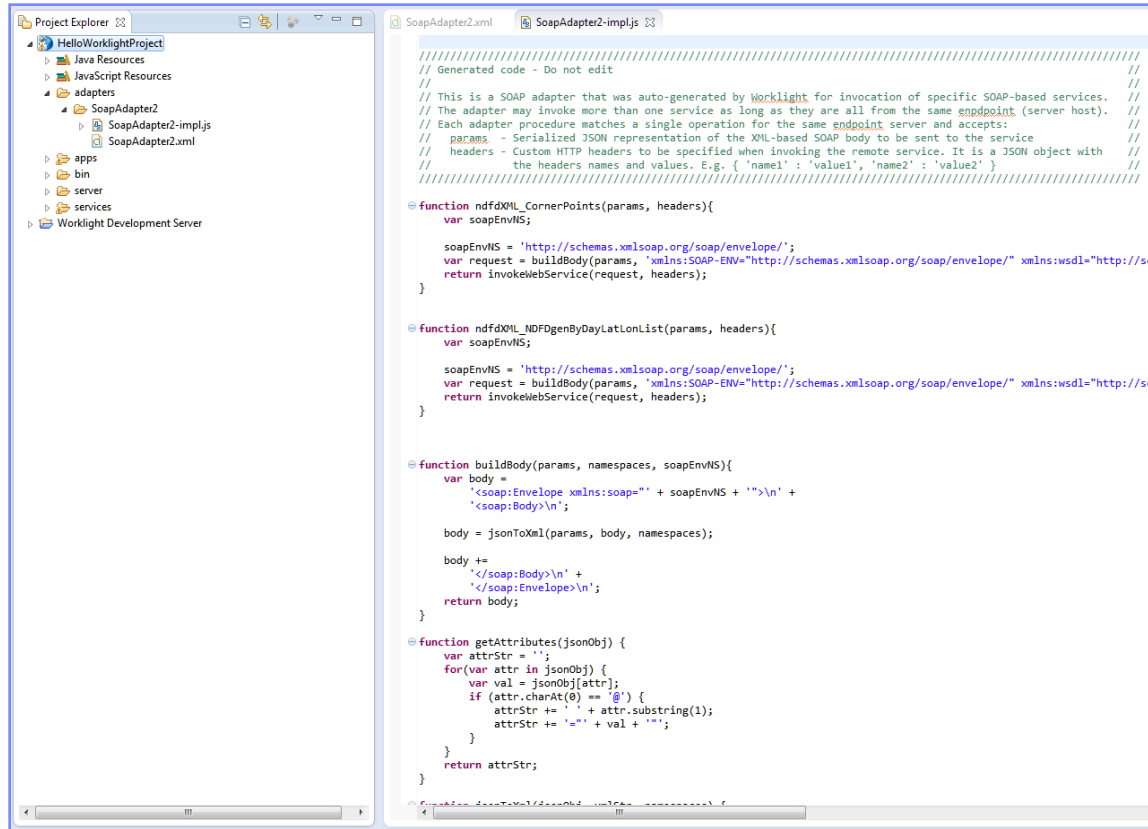
- 使用するサービスの場所を追加し、アダプターに追加するサービスを選択します。
- 追加するサービスごとに、この操作を繰り返します。サービスは同じアダプターに追加されます。





# バックエンド・サービスのディスカバリー – 続き

- 最終的に、以下のようなアダプターが自動生成されます。



The screenshot shows an IDE interface. On the left is the Project Explorer showing a project named 'HelloWorklightProject' with a folder 'adapters' containing 'SoapAdapter2' and 'SoapAdapter2-impl.js'. The main editor displays the content of 'SoapAdapter2-impl.js', which is auto-generated code for a SOAP adapter. The code includes comments explaining its purpose and structure, and defines several functions: 'ndfdXML\_CornerPoints', 'ndfdXML\_NDFDgenByDayLatLonList', 'buildBody', and 'getAttributes'.

```
// Generated code - Do not edit
//
// This is a SOAP adapter that was auto-generated by Worklight for invocation of specific SOAP-based services.
// The adapter may invoke more than one service as long as they are all from the same endpoint (server/host).
// Each adapter procedure matches a single operation for the same endpoint server and accepts:
//   params - Serialized JSON representation of the XML-based SOAP body to be sent to the service
//   headers - Custom HTTP headers to be specified when invoking the remote service. It is a JSON object with
//             the headers names and values. E.g. { 'name1': 'value1', 'name2': 'value2' }
//
// =====
@function ndfdXML_CornerPoints(params, headers){
    var soapEnvNS;

    soapEnvNS = 'http://schemas.xmlsoap.org/soap/envelope/';
    var request = buildBody(params, 'xmlns:SOAP-ENV=' + soapEnvNS);
    return invokeWebService(request, headers);
}

@function ndfdXML_NDFDgenByDayLatLonList(params, headers){
    var soapEnvNS;

    soapEnvNS = 'http://schemas.xmlsoap.org/soap/envelope/';
    var request = buildBody(params, 'xmlns:SOAP-ENV=' + soapEnvNS);
    return invokeWebService(request, headers);
}

@function buildBody(params, namespaces, soapEnvNS){
    var body =
        '<soap:Envelope xmlns:soap="' + soapEnvNS + '">\n' +
        '<soap:Body>\n';

    body = jsonToXml(params, body, namespaces);

    body +=
        '</soap:Body>\n' +
        '</soap:Envelope>\n';
    return body;
}

@function getAttributes(jsonObj) {
    var attrStr = '';
    for(var attr in jsonObj) {
        var val = jsonObj[attr];
        if (attr.charAt(0) == '@') {
            attrStr += '@' + attr.substring(1);
            attrStr += '=' + val + ';';
        }
    }
    return attrStr;
}
```

# アジェンダ

- HTTP アダプターとは
- HTTP アダプターを使用するための準備
- アダプターの作成
- SOAP の使用
  - SOAP ベースのサービス要求の作成
  - サービス要求の呼び出し
- バックエンド・サービスのディスカバリー
- 演習

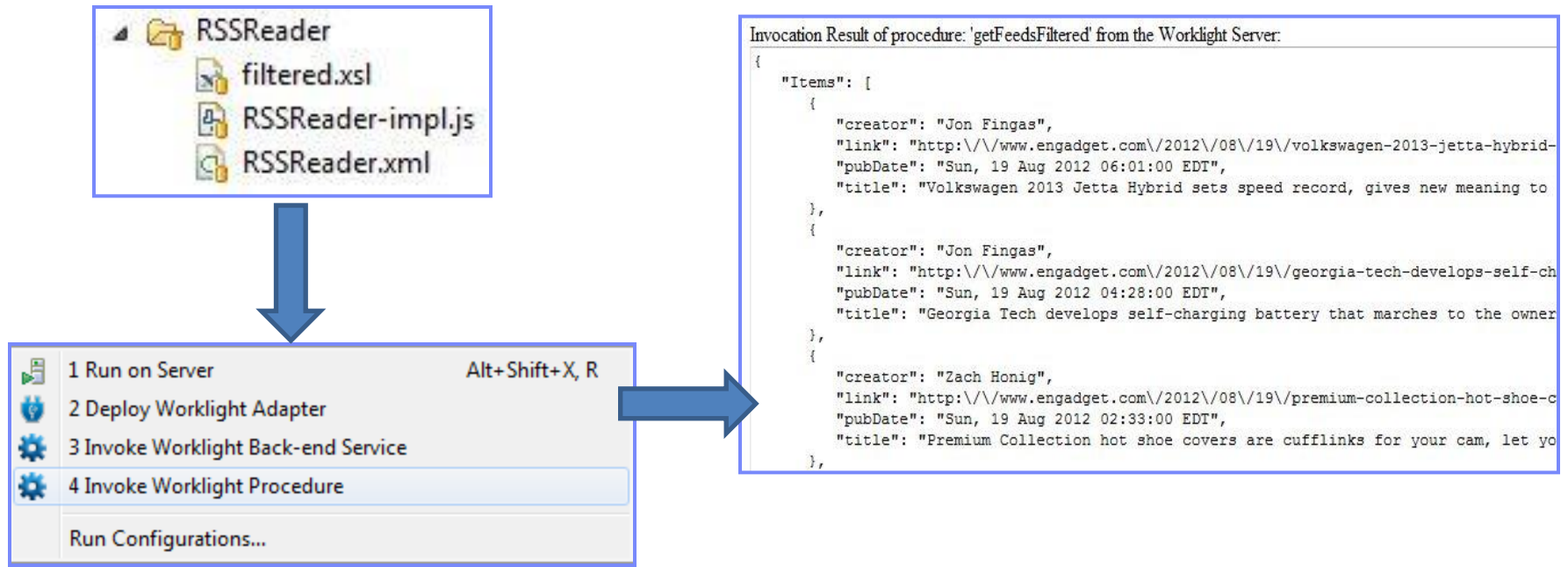
# 演習

## Engadget RSS リーダー

- **RSSReader** という名前の Worklight HTTP アダプターを作成します。
- <http://engadget.com/rss.xml> で Engadget RSS フィードに接続します。
- RSS フィードを取得する `getFeeds` プロシージャラーを宣言して実装します。
- `getFeedsFiltered` プロシージャラーを宣言して実装します。これは、`getFeeds` プロシージャラーと同様に機能しますが、`title`、`creator`、`pubDate`、および `link` フィールドのみを返します。
- アダプターをデプロイし、このモジュールの説明に従って Worklight Studio を使用してプロシージャラーをテストします。

# 演習 解答

- このトレーニング・モジュールのサンプルは、IBM Worklight 文書 Web サイト (<http://www.ibm.com/mobile-docs>) の「入門」ページにあります。



# 確認テスト

- HTTP アダプターはどのような目的で使用できますか。
  - RESTful サービスとの連携
  - SOAP サービスとの連携
  - GET 要求および POST 要求の発行
  - 上記のものすべて
- HTTP アダプターはどのような形式のデータを取得して自動的に解析することができますか。
  - XML
  - JSON
  - プレーン・テキスト
  - 上記のものすべて
- HTTP アダプターは非標準の HTTP ポートで使用できますか。
  - HTTP にはポート 80、HTTPS にはポート 443 を使用しなければなりません。
  - HTTP には任意のポートを使用できますが、HTTPS に使用できるのはポート 443 に限られます。
  - HTTP にはポート 80 を使用する必要がありますが、HTTPS には任意のポートを使用できます。
  - HTTP と HTTPS の両方に任意のポートを使用できます。

- HTTP アダプターはどのような目的で使用できますか。
  - RESTful サービスとの連携
  - SOAP サービスとの連携
  - GET 要求および POST 要求の発行
  - 上記のものすべて
- HTTP アダプターはどのような形式のデータを取得して自動的に解析することができますか。
  - XML
  - JSON
  - プレーン・テキスト
  - 上記のものすべて
- HTTP アダプターは非標準の HTTP ポートで使用できますか。
  - HTTP にはポート 80、HTTPS にはポート 443 を使用しなければなりません。
  - HTTP には任意のポートを使用できますが、HTTPS に使用できるのはポート 443 に限られます。
  - HTTP にはポート 80 を使用する必要がありますが、HTTPS には任意のポートを使用できます。
  - HTTP と HTTPS の両方に任意のポートを使用できます。

# 特記事項

- これらの資料は、以下のご使用条件に同意いただける場合に限りご使用いただけます。
- 本書は米国 IBM が提供する製品およびサービスについて作成したものです。
- 本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、または サービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。
- IBM は、本書に記載されている内容に関して特許権（特許出願中のものを含む）を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。
  - 〒103-8510  
東京都中央区日本橋箱崎町19番21号  
日本アイ・ビー・エム株式会社  
法務・知的財産  
知的財産権ライセンス渉外
- 以下の保証は、国または地域の法律に沿わない場合は、適用されません。IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態を提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。
- この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。
- 本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。
- IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。
- 本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム（本プログラムを含む）との間で情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。
  - IBM Corporation  
Dept F6, Bldg 1  
294 Route 100  
Somers NY 10589-3216  
USA

- 本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。
- 本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。
- IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確認できません。IBM 以外の製品の性能に関する質問は、それらの製品の供給者にお問い合わせください。
- 著作権使用許諾:**
  - 本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。
  - それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生的創作物にも、次のように、著作権表示を入れていただく必要があります。
    - © (お客様の会社名) (西暦年) このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。 © Copyright IBM Corp. \_年を入れる\_ All rights reserved.

## プライバシー・ポリシーの考慮事項

- サービス・ソリューションとしてのソフトウェアも含めた IBM ソフトウェア製品（「ソフトウェア・オファリング」）では、製品の使用に関する情報の収集、エンド・ユーザーの使用感の向上、エンド・ユーザーとの対話またはその他の目的のために、Cookie ははじめさまざまなテクノロジーを使用することがあります。多くの場合、ソフトウェア・オファリングにより個人情報が収集されることはありません。IBM の「ソフトウェア・オファリング」の一部には、個人情報を収集できる機能を持つものがあります。ご使用の「ソフトウェア・オファリング」が、これらの Cookie およびそれに類するテクノロジーを通じてお客様による個人情報の収集を可能にする場合、以下の具体的事項を確認ください。
- このソフトウェア・オファリングは、展開される構成に応じて、（アプリケーション・サーバーが生成する）セッション情報を収集するセッションごとの Cookie を使用する場合があります。これらの Cookie は個人情報を含まず、セッション管理のために要求されるものです。加えて、匿名ユーザーの認識および管理のために持続的な Cookie が無作為に生成される場合があります。これらの Cookie も個人情報を含まず、要求されるものです。
- この「ソフトウェア・オファリング」が Cookie およびさまざまなテクノロジーを使用してエンド・ユーザーから個人を特定できる情報を収集する機能を提供する場合、お客様は、このような情報を収集するにあたって適用される法律、ガイドライン等を遵守する必要があります。これには、エンドユーザーへの通知や同意の要求も含まれますがそれらには限られません。このような目的での Cookie を含む様々なテクノロジーの使用の詳細については、IBM の『IBM オンラインでのプライバシー・ステートメント』(<http://www.ibm.com/privacy/details/jp/ja/>) の「クッキー、ウェブ・ビーコン、その他のテクノロジー」および『IBM Software Products and Software-as-a-Service Privacy Statement』(<http://www.ibm.com/software/info/product-privacy/>) を参照してください。

# サポートおよびコメント

- IBM Worklight の一連の文書、トレーニング資料、および質問をポストできるオンライン・フォーラムはすべて、次の IBM Web サイトからご覧になれます。
  - <http://www.ibm.com/mobile-docs>
- サポート
  - ソフトウェア・サブスクリプション & サポート (ソフトウェア・メンテナンスと呼ばれる場合もあります) は、パスポート・アドバンテージおよびパスポート・アドバンテージ・エクスプレスから購入されたライセンスに含まれています。International Passport Advantage Agreement および IBM International Passport Advantage Express Agreement の追加情報については、次のパスポート・アドバンテージ Web サイトを参照してください。
    - <http://www.ibm.com/software/passportadvantage>
  - ソフトウェア・サブスクリプション & サポートが有効になっている場合、IBM は、インストールおよび使用法 (ハウツー) に関する短期間の FAQ に対するサポートや、コード関連の質問に対するサポートを提供します。詳しくは、次の IBM ソフトウェア・サポート・ハンドブックを参照してください。
    - <http://www.ibm.com/support/handbook>
- ご意見
  - 本資料に関するご意見をお寄せください。本資料の具体的な誤りや欠落、正確性、編成、題材、または完成度に関するご意見をお寄せください。お寄せいただくご意見は、本マニュアルまたは製品の情報、およびその情報の提示方法に関するもののみとしてください。
  - 製品の技術的な質問および情報、および価格については、担当の IBM 営業所、IBM ビジネス・パートナー、または認定リマーカーターにお問い合わせください。
  - IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。IBM またはいかなる組織も、お客様から提示された問題についてご連絡を差し上げる場合にのみ、お客様が提供する個人情報を使用するものとします。
  - どうぞよろしくお願いいたします。
  - 次の IBM Worklight Developer Edition サポート・コミュニティにご意見をお寄せください。
    - <https://www.ibm.com/developerworks/mobile/worklight/connect.html>
  - IBM からの回答を希望される場合は、以下の情報をご連絡ください。
    - 氏名
    - 住所
    - 企業または組織
    - 電話番号
    - Eメール・アドレス



ありがとうございました

