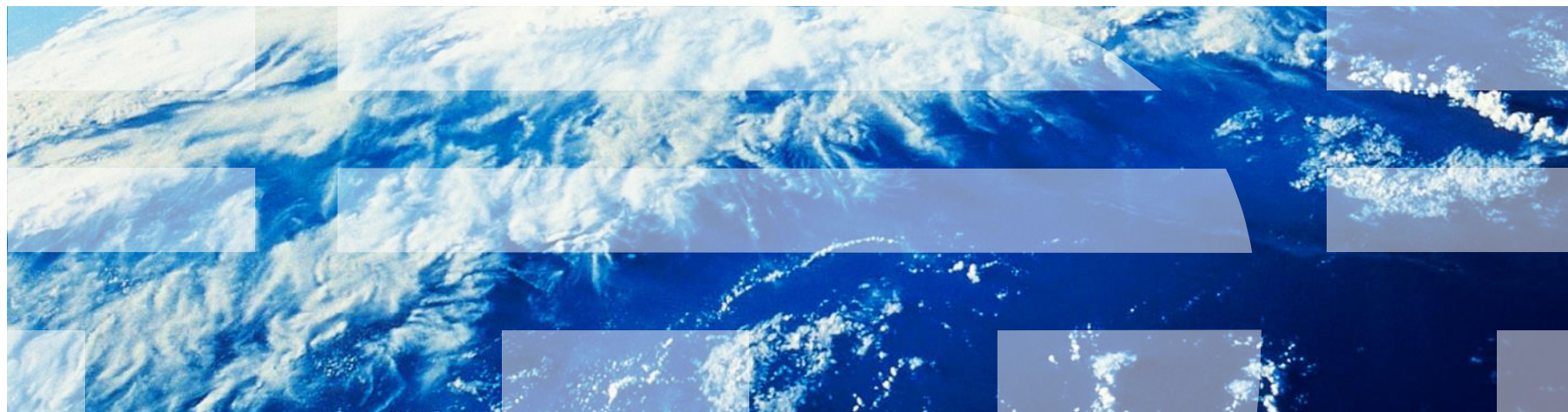


IBM Worklight V6.1.0 **入門**

アダプターの高度な使用方法とマッシュアップ



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

- サーバー・サイド・スクリプティング
 - 必要な理由
- Worklight サーバー・サイド API
 - プロシーチャーの呼び出し
 - サーバー構成の取得
 - ロギング
- データ・マッシュアップの例
 - Yahoo! Weather RSS データによる DB データの増補
- 演習

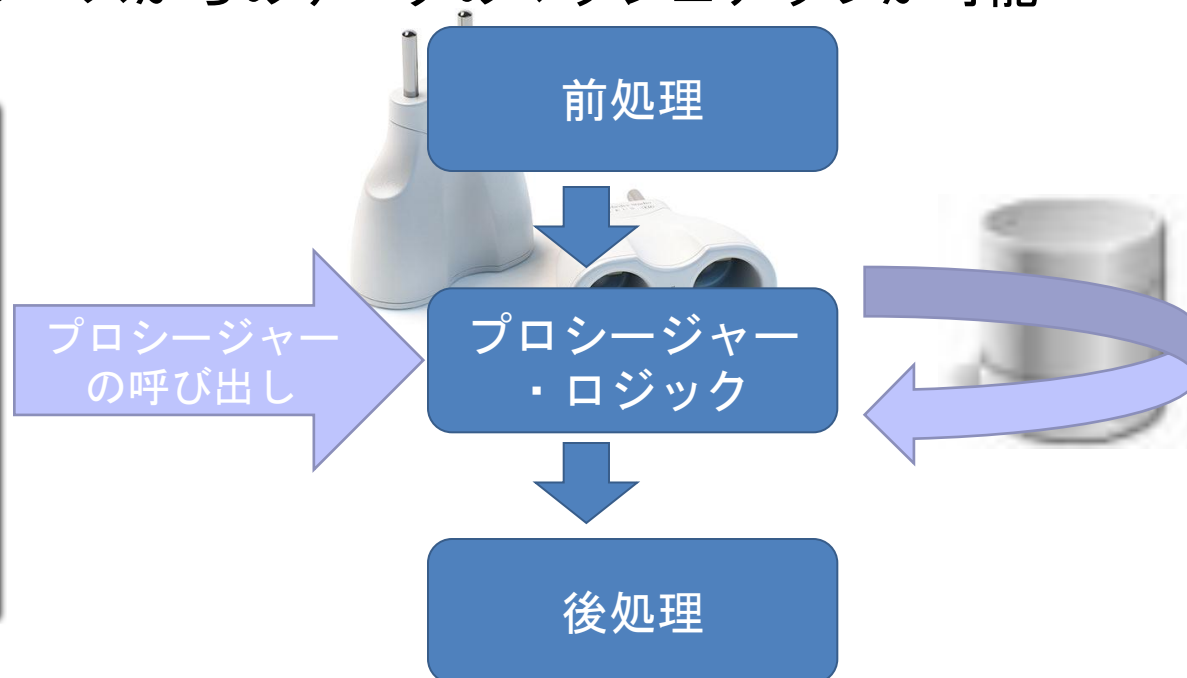
サーバー・サイド・スクリプティング 必要な理由

- アダプター機能の強化
 - 前処理ロジックと後処理ロジック
 - すべての処理が1回のサーバー呼び出しと1回のトランザクションのみで実行可能
 - 異なるデータ・ソースからのデータのマッシュアップが可能



Symbol	Price	Chg	% Chg
.SPX	1194.37	7.93	0.67
.DJI	10997.35	70.28	0.64
ORCL	26.13	0.05	0.19
AAPL	241.79	0.00	0.00
MOT	7.39	0.00	0.00
VRSN	26.84	0.00	0.00
MSFT	30.34	0.00	0.00
SNE	36.81	0.00	0.00
NOK	15.01	0.00	0.00

Quotes delayed at least 20 minutes
Fidelity Brokerage Services. Member
NYSE, SIPC



アジェンダ

- サーバー・サイド・スクリプティング
 - 必要な理由
- Worklight サーバー・サイド API
 - プロシーチャーの呼び出し
 - サーバー構成の取得
 - ロギング
- データ・マッシュアップの例
 - Yahoo! Weather RSS データによる DB データの増補
- 演習

Worklight® サーバー・サイド API

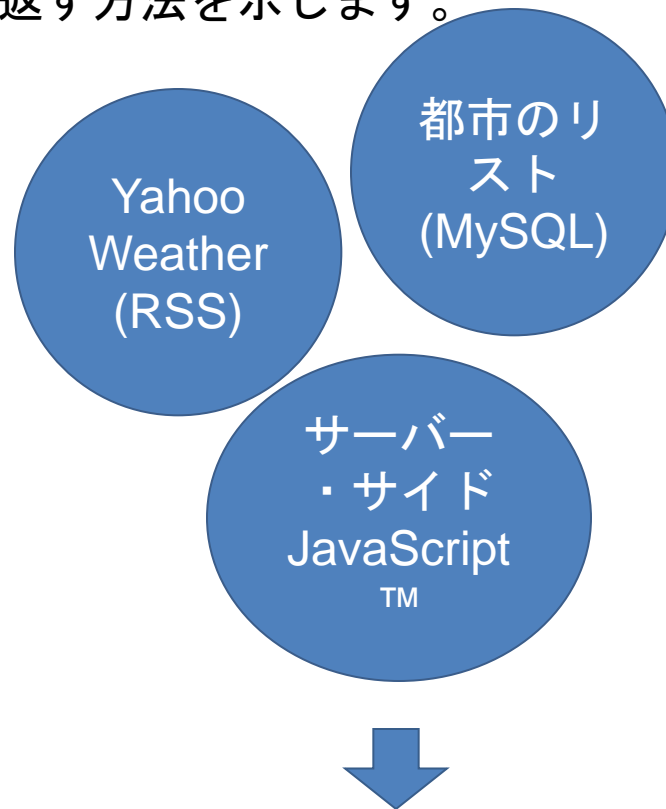
- **WL.Server.InvokeProcedure (invocationData)**
 - 現行のアダプターまたは別のアダプターからのプロシージャ呼び出し
 - クライアント・サイド `WL.Client.invokeProcedure` と同じ構文
- **WL.Logger.debug (msg)**
 - コンソールへのメッセージ出力
 - デバッグ用に使用
- **WL.Server.configuration オブジェクト**
 - `worklight.properties` ファイルで定義されているすべてのサーバー・プロパティを含むマップ。
 - 構文
 - `WL.Server.configuration["property-name"]`
 - 例
 - `var addr = WL.Server.configuration["local.IPAddress"];`

アジェンダ

- サーバー・サイド・スクリプティング
 - 必要な理由
- Worklight サーバー・サイド API
 - プロシーチャーの呼び出し
 - サーバー構成の取得
 - ロギング
- データ・マッシュアップの例
 - Yahoo! Weather RSS データによる DB データの増補
- 演習

データ・マッシュアップの例

- この例では、2つのデータ・ソースのデータをマッシュアップする方法およびデータ・ストリームを単一の `invocationResult` オブジェクトとしてアプリケーションに返す方法を示します。



データ・マッシュアップの例 - 続き

- データ・マッシュアップは、任意の数のデータ・ソースについて、また、異なるアダプター・タイプにわたって同じ方法で実装することができます。
- 例えば、以下のソースからのデータをマッシュアップするとします。
 - **SQL:**
 - 天気の表から都市のリストを取り出します。
 - 結果には、世界のいくつかの都市のリスト、それらの都市の Yahoo! Weather ID、および簡単な説明が含まれます。
 - **HTTP:**
 - Yahoo! Weather サービスに接続します。
 - SQL を使用して取り出したそれぞれの都市について、更新された天気予報を取り出します。
- その後、マッシュアップされたデータがアプリケーションに返されて表示されます。

データ・マッシュアップの例 - 続き


SQL アダプター

- SQL アダプターを作成して、**SQLAdapter** という名前を付けます。
 - このアダプターには、3つのプロシージャが含まれます。
 - **getCitiesWeather**: パブリック・プロシージャであり、アダプター XML ファイルで宣言されます。アプリケーションから呼び出され、マッシュアップされたデータを返します。
 - **getCitiesList**: 内部 (プライベート) 関数であり、アダプター XML ファイルで宣言されません。都市のリストを SQL サーバーから取得するため、サーバー API を使用してアダプターから呼び出されます。
 - **getCityWeather**: 内部 (プライベート) 関数であり、アダプター XML ファイルで宣言されません。Yahoo! Weather ID をパラメーターとして使用して、**getYahooWeather** プロシージャを別のアダプター (HTTP) から呼び出します。
 -
 - 関数はサーバー・サイド JavaScript™ を使用して実装されます。

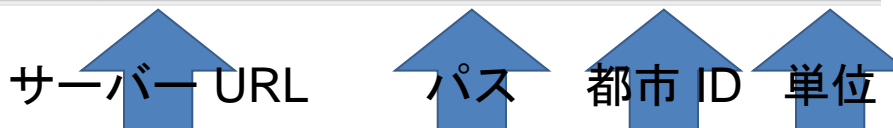
データ・マッシュアップの例 - 続き

HTTP アダプター

- HTTP アダプターを作成して、**HTTPWeather** という名前を付けます。
- このアダプターは Yahoo! Weather RSS フィード (<http://weather.yahooapis.com/forecastrss>) に接続します。
- アダプターには **getYahooWeather** という名前のプロシージャが 1 つだけあります。
- HTTP 要求には以下の 2 つのパラメーターがあります。
 - w – **世界における位置を表す ID** (Yahoo! で使用される都市 ID)
 - u – 単位を表し、摂氏の場合は「c」、華氏の場合は「f」
- 完全な要求 URL の例:

 <http://weather.yahooapis.com/forecastrss?w=2442047&u=c>

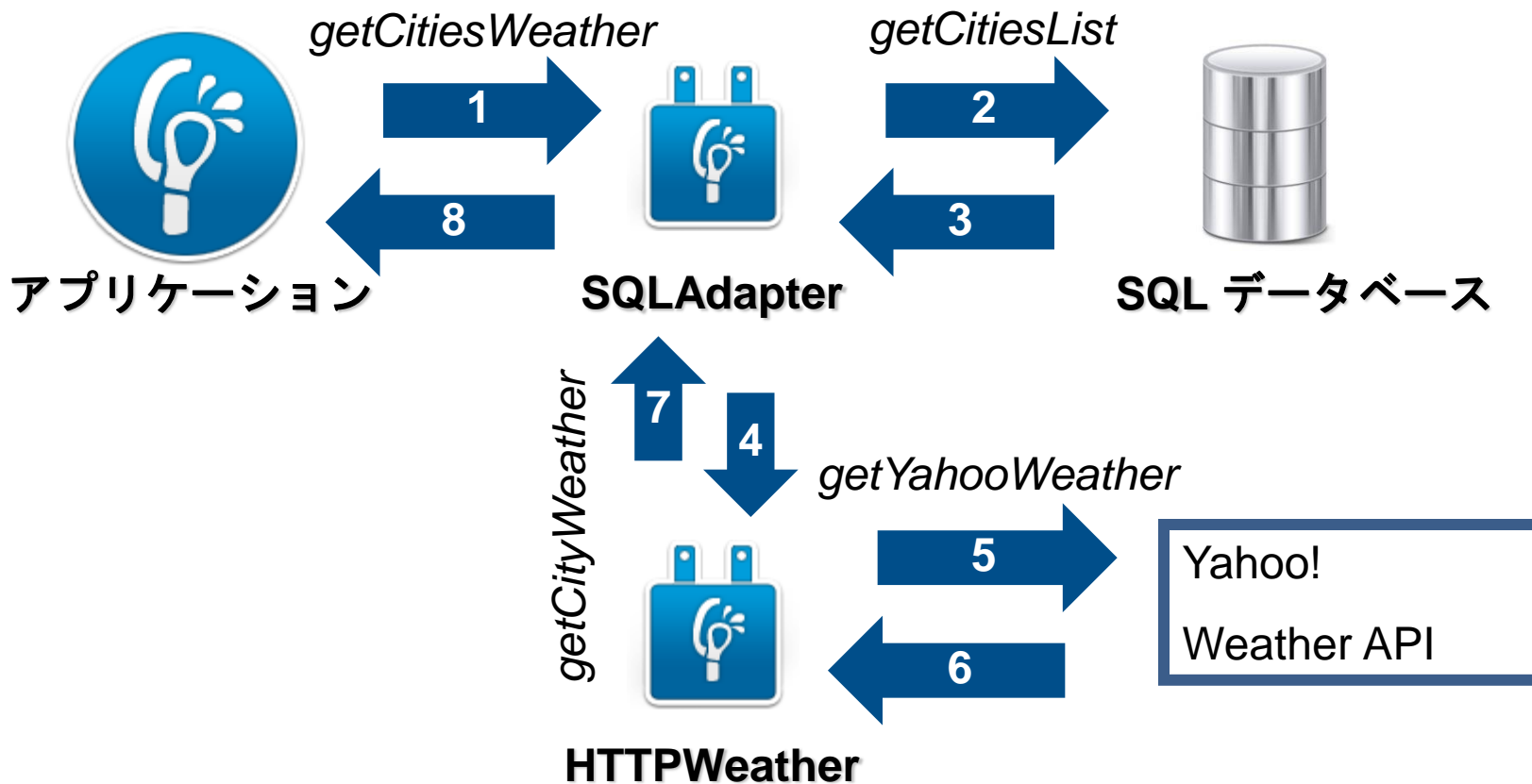
サーバー URL パス 都市 ID 単位



- 都市 ID パラメーターはプロシージャの呼び出し側によって指定されます。
- 単位パラメーターはハードコーディングされます。

データ・マッシュアップの例 - 続き

マッシュアップ・スキーマ



データ・マッシュアップの例 - 続き

マッシュアップ・スキーマ

アプリケーションが `getCitiesWeather` プロシージャを `SQLAdapter` から呼び出す

`SQLAdapter` が都市のリストを SQL データベースから取得する

`SQLAdapter` が都市 ID をパラメーターとして使用して `getYahooWeather` を `HTTPWeather` アダプターから呼び出す

`HTTPWeather` アダプターが気象データを Yahoo! Weather API から取得して `SQLAdapter` に返す

`SQLAdapter` がマッシュアップされたデータをアプリケーションに返す

都市ごとに
繰り返し実行

データ・マッシュアップの例 - 続き

サーバー・サイド JavaScript

- **getCitiesWeather** プロシーチャーを SQLAdapter XML ファイルで宣言します。
このプロシーチャーの機能は、2つの異なるアダプターからデータを取得して結果を1つのデータ・オブジェクトにマッシュアップすることです。
- **getCitiesList** 関数を SQLAdapter の .js ファイルで作成し、それを使用して都市のリストを SQL データベースから取得して返します。

```
var getCitiesListStatement =
    WL.Server.createStatement("select city, identifier, summary from weather;");
function getCitiesList() {
    return WL.Server.invokeSQLStatement({
        preparedStatement : getCitiesListStatement,
        parameters : []
    });
}
```

- 取得されたデータには以下のフィールドから構成される都市のリストが含まれています。
 - **city**: 都市名
 - **identifier**: Yahoo! Weather API 用の都市 ID
 - **summary**: 都市の簡単な説明

データ・マッシュアップの例 - 続き

サーバー・サイド JavaScript

- **getCityWeather** 関数を SQLAdapter の .js ファイルで作成し、それを使用して気象データを HTTPWeather アダプターから取得して返します。

```
function getCityWeather(woeid) {  
    return WL.Server.invokeProcedure({  
        adapter : 'HTTPWeather',  
        procedure : 'getYahooWeather',  
        parameters : [woeid]  
    });  
}
```

データ・マッシュアップの例 - 続き

サーバー・サイド JavaScript

- **getYahooWeather** プロシージャラーを HTTPWeather の XML ファイルで宣言します。
- このプロシージャラーのタスクは、指定された都市 ID をパラメーターとして使用して気象データを取得することです。
- **getYahooWeather** 関数を HTTPWeather の .js ファイルで作成し、その関数を使用して、Yahoo! Weather API で気象データを取得して返します。

```
function getYahooWeather(woeid) {  
  
    var input = {  
        method : 'get',  
        returnedContentType : 'xml',  
        path : 'forecastrss',  
        parameters : {  
            'w' : woeid,  
            'u' : 'c' //celcius  
        }  
    };  
  
    return WL.Server.invokeHttp(input);  
}
```


データ・マッシュアップの例 - 続き

サーバー・サイド JavaScript

- SQLAdapter で、事前取得された都市のリストから都市 ID を取り出し、それを使用して HTTPWeather アダプターの **getYahooWeather** プロシーチャーを呼び出します。
- この手順を都市ごとに繰り返し、受け取ったデータを city オブジェクトに追加します。

```
function getCitiesWeather(){  
    var cityList = getCitiesList();  
    for (var i = 0; i < cityList.resultSet.length; i++) {  
        var yahooWeatherData = getCityWeather(cityList.resultSet[i].identifier);  
        cityList.resultSet[i].weather = yahooWeatherData;  
    }  
    return cityList;  
}
```

- マッシュアップされたデータをアプリケーションに返します。

データ・マッシュアップの例 - 続き

結果の確認

Invocation Result from the WorkLight Server:

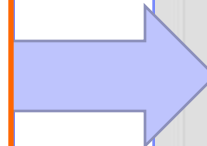
```

{
  "isSuccessful": true,
  "resultSet": [
    {
      "city": "New York",
      "identifier": "2459115",
      "summary": "New York City, which is geographically the largest city in the state history as a gateway for immigration to the United States and its status as a financial According to the U.S. Department of Commerce, it is also a destination of choice for ma 17th century Duke of York, James Stuart, future James II and VII of England and Scotlan
      "weather": {
        "errors": [],
        "info": [],
        "isSuccessful": true,
        "rss": {
          "channel": {
            "astronomy": {
              "sunrise": "7:18 am",
              "sunset": "4:32 pm"
            },
            "atmosphere": {
              "humidity": "83",
              "pressure": "982.05",
              "rising": "0",
              "visibility": "2.41"
            },
            "description": "Yahoo! Weather for New York, NY",
            "image": {
              "height": "18",
              "link": "http://weather.yahoo.com",
              "title": "Yahoo! Weather",
              "url": "http://l.yimg.com/a/i/brand/purplelogo//uh/us/news-wea.gif",
              "width": "142"
            }
          }
        }
      }
    }
  ]
}

```

SQL データ

**Yahoo!
Weather**



CityWeather

Select city:



Current Conditions:
Cloudy, -5 C

Forecast:
Mon - Blizzard. High: -3 Low: -6
Tue - Mostly Sunny. High: 2 Low: -4

[Full Forecast at Yahoo! Weather](#)

(provided by [The Weather Channel](#))

New York City, which is geographically the largest city in the state and most populous in the United States, is known for its history as a gateway for immigration to the United States and its status a...

アジェンダ

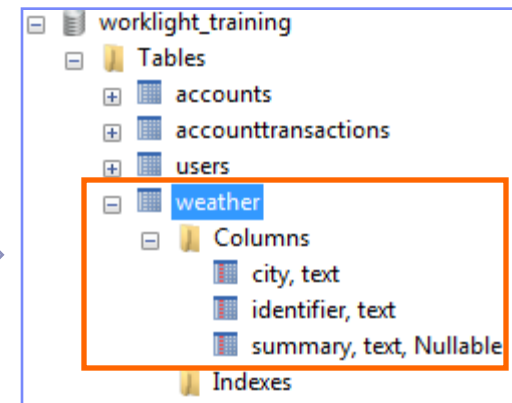
- サーバー・サイド・スクリプティング
 - 必要な理由
- Worklight サーバー・サイド API
 - プロシーチャーの呼び出し
 - サーバー構成の取得
 - ロギング
- データ・マッシュアップの例
 - Yahoo! Weather RSS データによる DB データの増補
- 演習

演習: 準備

CityWeather アプリケーション

- データベースを作成します。
 - ローカル MySQL サーバーの ¥server フォルダにある worklightTraining.sql ファイルを実行して、トレーニング・データベースを作成します。

```
/*!40101 SET NAMES utf8 */;  
/*!40101 SET SQL_MODE='***/;  
  
CREATE DATABASE IF NOT EXISTS `WorkLight Training`;  
  
USE `WorkLight Training`;  
  
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,  
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VA  
/*!Table structure for table `accounts` */  
  
DROP TABLE IF EXISTS `accounts`;  
  
CREATE TABLE `accounts` (  
  `accountId` VARCHAR(15) NOT NULL,  
  `userId` VARCHAR(100) NOT NULL DEFAULT '',  
  `accountType` VARCHAR(50) NOT NULL
```



演習

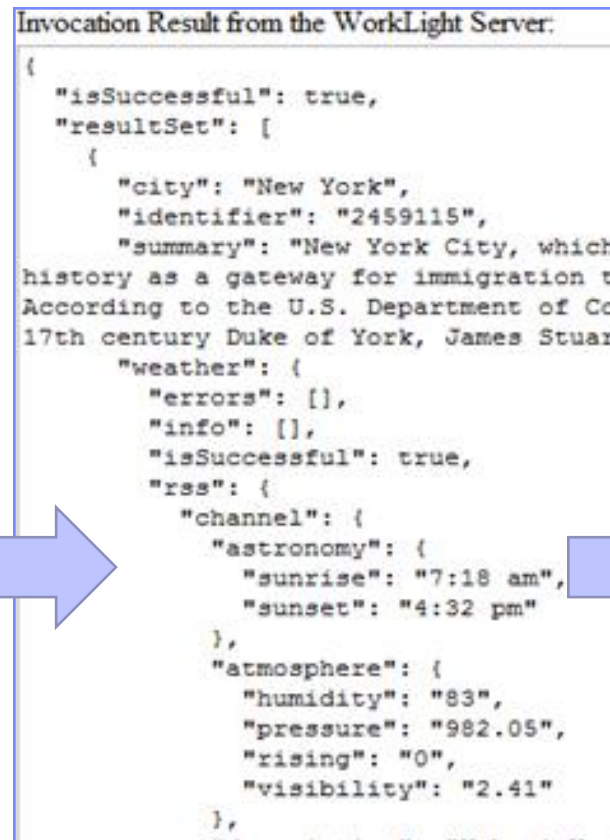
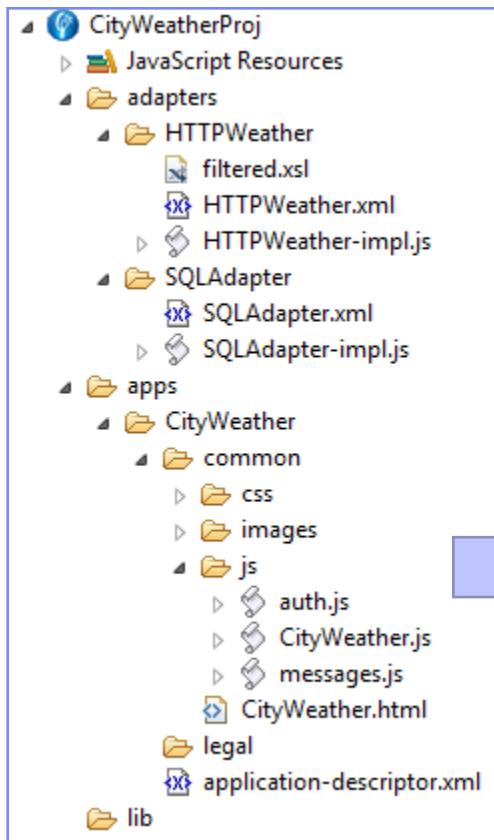
- CityWeather アプリケーションを作成して、このモジュールで示している例を実装します。
- 2つのアダプター (SQL および HTTP) を実装する必要があります。
- SQLAdapter には JDBC コネクター・ドライバーが必要です。
 - JDBC コネクター・ドライバーは開発者が別個にダウンロードし、Worklight プロジェクトの lib¥ フォルダに追加する必要があります。
 - JDBC コネクター・ドライバーは MySQL、Oracle、および DB2 のそれぞれの Web サイトでダウンロードすることができます。
 - この演習では、MySQL 版をダウンロードします。
- SQLAdapter の XML ファイルの connectivity セクションで、データベースの接続情報を以下のように指定します。

```
<connectivity>
  <connectionPolicy xsi:type="sql:SQLConnectionPolicy">
    <dataSourceDefinition>
      <driverClass>com.mysql.jdbc.Driver</driverClass>
      <url>jdbc:mysql://127.0.0.1:3306/worklight_training</url>
      <user>Worklight</user>
      <password>Worklight</password>
    </dataSourceDefinition>
  </connectionPolicy>
  <loadConstraints maxConcurrentConnectionsPerNode="5" />
</connectivity>
```

- アプリケーションは、開始されると、SQL アダプターを使用して SQL データベースに接続し、都市のリストを気象情報テーブルから取得します。
- 都市のリストが受け取られると、SQL アダプターが HTTP アダプターを使用して Yahoo! Weather API に接続し、リストのそれぞれの都市の気象データを取得します。
- マッシュアップされたデータをアプリケーションが受け取って表示します。
- 注: このアプリケーションではアダプターを 1 回だけ呼び出します。すべてのマッシュアップ・ロジックはサーバー・サイド API によって実装されます。

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。



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、または認定リマーケットア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

