

IBM Worklight V6.1.0 **入門**

JSONStore – JSONStore の一般的な使用法



商標

- IBM、IBM ロゴおよび 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、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。

概要

- このモジュールは、次の 2 つの部分で構成されています。
 - 基本的な使用法: ローカルの JSONStore コレクションで実行できる基本的なタスク (追加 や 検索 など)。
 - 高度な使用法: データ・ストレージに対するビジネス要件に応じた、役立つと思われる高度なタスク (セキュリティー など)。
- このモジュールで使用されているコードのサンプルは、意図的に短くしてあります。
 - 完全なサンプルが、このモジュールに付属するサンプル・アプリケーションで使用できます。
 - このモジュールで使用されている画面キャプチャーは、サンプル・アプリケーションから取得されたものです。

アジェンダ

- 基本的な使用法

- 初期化
- 取得
- 追加
- 検索
- 置換
- 削除
- コレクションの削除
- 破棄

- 高度な使用法

- セキュリティー
- 複数ユーザーのサポート
- Worklight アダプターの統合
- 拡張

初期化 – 概説

- 1 つ以上の JSONStore コレクションを開始するには `init` を使用します。
- コレクションの開始またはプロビジョンは、コレクションとドキュメントが含まれる永続ストレージが作成されることを意味します (存在しない場合)。
- 永続ストレージが暗号化され、正しいパスワードが渡されると、そのデータにアクセスできるようにするための、セキュリティー上必要な手順が実行されます。
- 初期化時に有効化できるオプション・フィーチャーについては、このモジュールの後半にある『セキュリティー』、『複数ユーザーのサポート』、および『Worklight アダプターの統合』を参照してください。

初期化 - 例

```
var collectionName = 'people';

//Object that defines all the collections
var collections = {};

//Object that defines the 'people' collection
collections[collectionName] = {};

//Object that defines the Search Fields for the 'people' collection
collections[collectionName].searchFields = {name: 'string', age: 'integer'}

WL.JSONStore.init(collections)
.then(function () {
    //handle success
})
.fail(function (errorObject) {
    //handle failure
});
```

Initialize (Login/Open)

取得 - 概説

- コレクションへのアクセス機能を作成するには、`get` を使用します。
- `get` を呼び出す前に、`init` を呼び出す必要があります。そうしない場合、`get` の結果が未定義になります。

取得 - 例

```
var collectionName = 'people';  
var people = WL.JSONStore.get(collectionName);
```

- 変数 **people** を使用して、**people** コレクションに対する以下のような操作を実行できます。
 - add
 - find
 - replace

追加 – 概説

- コレクション内にデータをドキュメントとして保管するには、add を使用します。

追加 - 例

```
var data = {name: 'carlos', age: 99};  
var collectionName = 'people';  
var options = {}; //default  
WL.JSONStore.get(collectionName)  
  .add(data, options)  
  .then(function () {  
    //handle success  
  })  
  .fail(function (errorObject) {  
    //handle failure  
  });
```

Enter Name	Enter Age
Add Data	

検索 - 概説

- 照会を使用してコレクション内のドキュメントを見つけるには、`find` を使用します。
- コレクション内のすべてのドキュメントを取り出すには、`findAll` を使用します。
- ドキュメントの固有 ID で検索するには、`findById` を使用します。
- `find` のデフォルトの動作は、ファジー検索を実行します。

检索 - 例

```
var query = {name: 'carlos'};
var collectionName = 'people';
var options = {
  exact: false, //default
  limit: 10 //returns a maximum of 10 documents, default: return every match
};
```

```
WL.JSONStore.get(collectionName)
```

```
.find(query, options)
```

```
.then(function (arrayResults) {
  //arrayResults = [{_id: 1, json: {name: 'carlos', age: 99}}]
})
```

```
.fail(function (errorObject) {
  //handle failure
});
```

Search Field	
Limit (optional)	Offset (optional)
Find By Name (Fuzzy Search)	
Find By Age (Exact Search)	
Find All	
Enter Id	Find (by id)

置換 - 概説

- コレクション内のドキュメントを変更するには、`replace` を使用します。
- 置換の実行に使用するフィールドは `_id` で、これはドキュメントの固有 ID です。

置換 - 例

```
var document = {_id: 1, json: {name:
  'carlitos', age: 99}};
var collectionName = 'people';
var options = {}; //default
```

```
WL.JSONStore.get(collectionName)
```

```
.replace(document, options)
```

```
.then(function () {
  //handle success
})
```

```
.fail(function (errorObject) {
  //handle failure
});
```

- この例では、ドキュメント `{_id: 1, json: {name: 'carlos', age: 99}}` がコレクション内にあるものと想定し、このドキュメントを名前が `'carlitos'` であるドキュメントに置換する方法を示しています。

Enter Name	Enter Age
Enter Id	Replace (by id)

削除 – 概説

- ドキュメントをコレクションから削除するには、`remove` を使用します。
- `push` が呼び出されるまで、ドキュメントはコレクションから消去されません。
 - 詳しくは、このモジュールの後半にある『**Worklight アダプターの統合**』セクションを参照してください。

削除 - 例

```
var query = {_id: 1};
var collectionName = 'people';
var options = {exact: true}; //default: false

WL.JSONStore.get(collectionName)

.remove(query, options)

.then(function () {
    //handle success
})

.fail(function (errorObject) {
    //handle failure
});
```

コレクションの削除 – 概説

- コレクション内に保管されているすべてのドキュメントを削除するには、`removeCollection` を使用します。
 - この操作は、データベース用語における、表のドロップと似ています。

コレクションの削除 – 例

```
var collectionName = 'people';

WL.JSONStore.get(collectionName)

.removeCollection()

.then(function () {
    //handle success
})

.fail(function (errorObject) {
    //handle failure
});
```

A button with a light gray gradient and rounded corners, containing the text "Remove Collection".

Remove Collection

破棄 – 概説

- 以下のデータを削除するには、`destroy` を使用します。
 - すべてのドキュメント
 - すべてのコレクション
 - すべてのストア
 - このモジュールの後半にある『複数ユーザーのサポート』を参照してください。
 - すべての JSONStore メタデータおよびセキュリティー成果物
 - このモジュールの後半にある『セキュリティー』を参照してください。

破棄 - 例

```
var collectionName = 'people';
```

```
WL.JSONStore.destroy()
```

```
.then(function () {  
    //handle success  
})
```

```
.fail(function (errorObject) {  
    //handle failure  
});
```



Destroy Everything

アジェンダ

- 基本的な使用法
 - 初期化
 - 取得
 - 追加
 - 検索
 - 置換
 - 削除
 - コレクションの削除
 - 破棄
- 高度な使用法
 - セキュリティー
 - 複数ユーザーのサポート
 - Worklight アダプターの統合
 - 拡張

セキュリティー – 概説

- パスワードを `init` 関数に渡すことにより、ストア内のすべてのコレクションを保護することができます。
- パスワードを渡さないと、ストア内のすべてのコレクションにあるドキュメントが暗号化されません。
- データ暗号化は Android 環境および iOS 環境でのみ使用可能です。
- セキュリティー・メタデータの一部は、以下の場所に保管されます。
 - iOS では Keychain
 - Android では SharedPreferences
- ストアは 256 ビットの Advanced Encryption Standard (AES) 鍵で暗号化されます。すべての鍵は Password-Based Key Derivation Function 2 (PBKDF2) により強化されています。
- `closeAll` を使用して、`init` を再度呼び出すまですべてのコレクションへのアクセスをロックします。
 - `init` をログイン関数と考えると、`closeAll` はそれに対応するログアウト関数と考えることができます。
- `changePassword` を使用して、パスワードを変更します。

セキュリティー 例

```
var collections = {
  people: {
    searchFields: {name: 'string'}
  }
};
```

```
var options = {
  password: '123'
};
```

```
WL.JSONStore.init(collections, options)
```

```
.then(function () {
  //handle success
})
```

```
.fail(function (errorObject) {
  //handle failure
});
```

複数ユーザーのサポート – 概説

- 単一の Worklight アプリケーションに、異なるコレクションを含む複数のストアを作成できます。
- `init` 関数では、ユーザー名を含む options オブジェクトを使用することができます。
- ユーザー名を指定しない場合、デフォルトのユーザー名は **“jsonstore”** です。

複数ユーザーのサポート - 例

```
var collections = {  
  people: {  
    searchFields: {name: 'string'}  
  }  
};
```

```
var options = {  
  username: 'carlos'  
};
```

```
WL.JSONStore.init(collections, options)
```

```
.then(function () {  
  //handle success  
})
```

```
.fail(function (errorObject) {  
  //handle failure  
});
```

Worklight アダプターの統合 – 概説

- このセクションは、ユーザーが Worklight アダプターについて理解していることを前提とします。
- Worklight アダプターの統合は任意です。統合により、以下を行うことができます。
 - データをコレクションから Worklight アダプターに送信します。
 - Worklight アダプターからコレクションへデータを取得します。
- これらの目的を実現するために、**WL.Client.invokeProcedure** や、より高い柔軟性が必要な場合は **jQuery.ajax** などの関数を使用することができます。
- Worklight Studio では、JavaScript™ テンプレート・ファイルを作成するための新しいウィザードを使用できます。
 - これは、アダプターとの通信に使用したバックエンド・システムから選択される検索フィールドに基づいて作成されます。

アダプターの実装 - 例

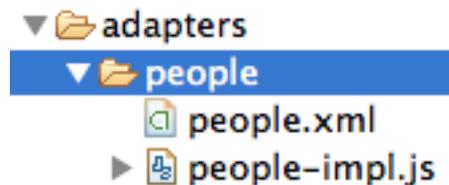
- まず、Worklight アダプターを作成し、データを取得、追加、置換、および削除するプロシージャー (**getPeople**、**addPerson**、**replacePerson**、および **removePerson** など) を定義します。
- 次に、これらのプロシージャーを実装します。
 - 分かりやすくするため、以下の例は、ハードコーディングされたデータとログのみを返します。
 - Worklight アダプターの作成方法について詳しくは、関連する IBM Worklight ユーザー文書 (http://pic.dhe.ibm.com/infocenter/wrklght/v6r1m0/topic/com.ibm.worklight.getstart.doc/start/c_gettingstarted.html) 内のカテゴリ 4 『Worklight サーバー・サイド開発』にあるモジュールを参照してください。

- ```
function getPeople () {
 return { peopleList : [{name: 'carlos', age: 100},
 {name: 'tim', age: 99}] };
}

function addPerson (data) {
 return WL.Logger.debug('Got data from
JSONStore to ADD: ' + data);
}

function replacePerson (data) {
 return WL.Logger.debug('Got data from
JSONStore to REPLACE: ' + data);
}

function removePerson (data) {
 return WL.Logger.debug('Got data from
JSONStore to REMOVE: ' + data);
}
```



## Worklight アダプターにリンクされたコレクションの初期化 – 例

- **init** を呼び出し、アダプター・メタデータ (アダプター名、プロシージャー名) を渡して people コレクションを開始します。
- この例で、キーは *peopleList* です。これは、`{name: 'carlos', age: 100}` および `{name: 'tim', age: 99}` を2つの個別のドキュメントとして保管することが目的であるためです。

```

var collections = {
 people: {
 searchFields: {
 name: 'string',
 age: 'integer'},
 //-- Start adapter metadata
 adapter : {
 name: 'people',
 add: 'addPerson',
 remove: 'removePerson',
 replace: 'replacePerson',
 load: {
 procedure: 'getPeople',
 params: [],
 key: 'peopleList'
 }
 }
 }
 //-- End adapter metadata
}; // (continued on the right column)

```

```

// (continued from the left column)

//Initialize
WL.JSONStore.init(collections)

.then(function () {
 //handle success
})

.fail(function (errorObject) {
 //handle failure
});

```

user (optional)

pass (optional)

Initialize (Login/Open)

## データを Worklight アダプターからロード - 例

- `load` が呼び出されるときに、JSONStore は、前に `init` に渡した、アダプターに関するメタデータ (*name* や *procedure* など) を使用してアダプターから取得するデータを決定し、最終的にはそのデータを保管します。

```
WL.JSONStore.get('people').load()
.then(function () {
 //handle success
})
.fail(function (errorObject) {
 //handle failure
});
```

Load Data From Adapter

## プッシュが必要な対象の取得 – 例

- `getPushRequired` を呼び出すと、「ダーティーなドキュメント」と呼ばれるドキュメントの配列が返されます。これは、バックエンド・システムには存在しないローカル変更が含まれるドキュメントを指します。
  - これらのドキュメントは、`push` が呼び出されたときに、Worklight アダプターに送信されます。

```
WL.JSONStore.get('people').getPushRequired()
 .then(function () {
 //handle success
 })
 .fail(function (errorObject) {
 //handle failure
 });
```

- JSONStore がドキュメントを「ダーティー」としてマーク付けするのを防止するには、`add`、`replace`、および `remove` にオプション `{push: false}` を渡します。

Get Push Required

## プッシュ - 例

- `push` は、変更されたドキュメントを正しい Worklight アダプター・プロシージャーに送信します (例えば、ローカルで追加されたドキュメントの場合は `addPerson` が呼び出されます)。
  - このメカニズムは、変更されたドキュメント、および `init` に渡されるアダプター・メタデータに関連する、最後のオペレーションに基づきます。

```
WL.JSONStore.get('people').push()
.then(function (res) {
 //handle success
 //res is an empty array if all documents reached the server
 //res is an array of error responses if some documents failed
 to reach the server
})
.fail(function (errorObject) {
 //handle failure
});
```

Push Changes to Adapter

## 拡張 - 概説

- 関数をコレクションのプロトタイプに追加することにより、コア API をニーズに合うように拡張するには、`enhance` を使用します。

## 拡張 - 例

- この例は、`enhance` を使用して、**keyvalue** コレクションで動作する関数 **getValue** を追加する方法を示しています。
  - 唯一のパラメーターとして **key** (ストリング) を使用し、単一の結果を返します。
  - このモジュールに関連するコード・サンプルで、**init** の呼び出しと、`enhance` を使用して **put** メソッドを取得する方法が示されています。

```
//Definition:
var collectionName = 'keyvalue';
WL.JSONStore.get(collectionName).enhance(
 'getValue', function (key) {
 var deferred = $.Deferred(),
 collection = this;

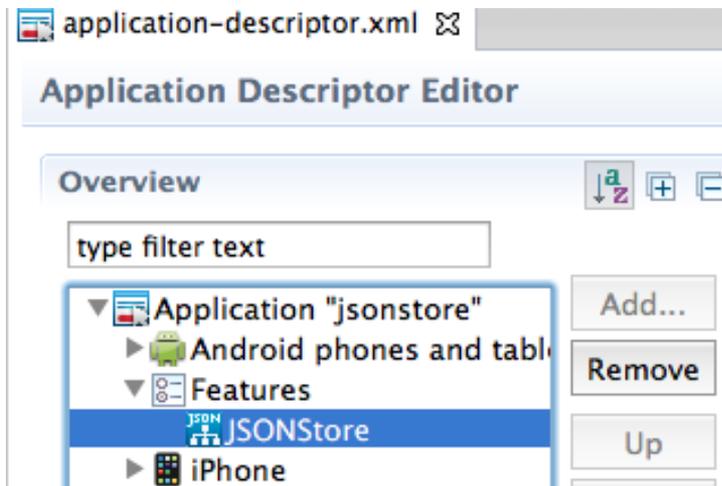
 //Do an exact search for the key
 collection.find({key: key}, {exact:
 true, limit: 1})
 .then(deferred.resolve,
 deferred.reject);
 return deferred.promise();
 }); // (continued on the right column)
```

```
// (continued from the left column)
//Usage:
var key = 'myKey';
WL.JSONStore.get(collectionName).getValue(
 key)
 .then(function (res) {
 //res contains an array of documents
 //with the results from the find
 })
 .fail(function () {
 //handle failure
 });
```



## サンプル

- このトレーニング・モジュールのサンプルは、IBM Worklight 文書 Web サイトの「入門」ページ (<http://www.ibm.com/mobile-docs>) にあります。
- このモジュールのサンプルを使用するには、IBM Worklight V6.0.0 Interim Fix (IF) 201307011413 以降をインストールする必要があります。
- アプリケーションで、JSONStore フィーチャーが有効になっていることを確認してください。



## 詳細情報

- JSONStore に関する詳細情報 (ファジー検索や Worklight Studio ウィザード、およびこのモジュールで紹介されている他のトピックなどに関する詳細情報) については、IBM Worklight ユーザー文書で JSONStore に関する資料を参照してください。
- 一般的なトピック:  
[http://pic.dhe.ibm.com/infocenter/wrklight/v6r1m0/topic/com.ibm.worklight.dev.doc/devref/c\\_jsonstore\\_overview.html](http://pic.dhe.ibm.com/infocenter/wrklight/v6r1m0/topic/com.ibm.worklight.dev.doc/devref/c_jsonstore_overview.html)
- インスタンスの関数:  
<http://pic.dhe.ibm.com/infocenter/wrklight/v6r1m0/topic/com.ibm.worklight.apiref.doc/html/refjavascript-client/html/JSONStoreInstance.html>
- 静的関数:  
<http://pic.dhe.ibm.com/infocenter/wrklight/v6r1m0/topic/com.ibm.worklight.apiref.doc/html/refjavascript-client/html/WL.JSONStore.html>

# 特記事項

- これらの資料は、以下のご使用条件に同意して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）との間での情報交換、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をお願いします。

## 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めめしたり、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。
  - どうぞよろしく願いたします。
  - 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

