



Version 29.0: Winter '14

Force.com REST API 開発者ガイド



最終更新日: 2013/11/26

目次

Force.com REST API の使用開始.....	1
第 1 章: Force.com REST API の概要.....	1
Force.com REST リソースの理解.....	2
圧縮の使用.....	3
REST での cURL の使用例.....	4
認証について.....	4
接続アプリケーションの定義.....	5
OAuth エンドポイントについて.....	5
Web サーバ OAuth 認証フローについて.....	6
ユーザエージェント OAuth 認証フローについて.....	10
ユーザ名パスワード OAuth 認証フローについて.....	13
OAuth 更新トークンプロセスについて.....	16
その他のリソースを見つける.....	17
第 2 章: クイックスタート.....	19
前提条件.....	20
ステップ 1: Salesforce Developer Edition 組織を取得する.....	20
ステップ 2: 認証を設定する.....	20
ステップ 3: cURL で HTTP 要求を送信する.....	23
ステップ 4: サンプルコードを実行する.....	24
REST リソースの使用.....	30
第 3 章: REST API リソースの使用.....	30
組織に関する情報の取得.....	31
使用可能な REST API バージョンをリストする.....	31
使用可能な REST リソースをリストする.....	32
オブジェクトのリストを取得する.....	32
オブジェクトメタデータの使用.....	33
オブジェクトのメタデータを取得する.....	33
オブジェクトの項目と他のメタデータを取得する.....	34
オブジェクトのメタデータの変更の取得.....	35
レコードの操作.....	36
レコードを作成する.....	36
レコードを更新する.....	37
レコードを削除する.....	38
レコードから項目値を取得する.....	38
外部 ID を使用してレコードを取得する.....	39
外部 ID を使用してレコードを挿入/更新 (Upsert) する.....	39

レコードから添付ファイルコンテンツを取得する.....	43
Blob データを挿入または更新する.....	43
特定の期間に削除されたレコードのリストの取得.....	47
特定の期間に更新されたレコードのリストの取得.....	48
検索とクエリの使用.....	49
SOQL クエリを実行する.....	49
削除された項目を含む SOQL クエリを実行する.....	50
文字列を検索する.....	51
デフォルトの検索範囲と検索順序の取得.....	53
オブジェクトの検索結果レイアウトの取得.....	53
最近参照した情報の操作.....	55
最近参照したレコードの表示.....	55
最近参照したデータとしてレコードをマーク.....	55
ユーザパスワードの管理.....	56
ユーザパスワードを管理する.....	56
REST API リファレンス.....	59
第 4 章: リファレンス.....	59
Versions.....	61
Resources by Version.....	62
Describe Global.....	62
sObject Basic Information.....	63
sObject Describe.....	63
sObject Get Deleted.....	64
sObject Get Updated.....	65
sObject Rows.....	67
sObject Rows by External ID.....	67
sObject Blob Retrieve.....	68
sObject CompactLayouts.....	69
sObject Layouts.....	70
sObject Quick Actions.....	72
sObject User Password.....	73
AppMenu.....	74
FlexiPage.....	77
Query.....	79
QueryAll.....	80
Quick Actions.....	81
Search.....	82
Search Scope and Order.....	83
Search Result Layouts.....	83
Recently Viewed Items.....	84
Themes.....	84

ヘッダー.....	86
割り当てルールヘッダー.....	87
Limit Info ヘッダー.....	87
状況コードとエラー応答.....	88

Force.com REST API の使用開始

第 1 章

Force.com REST API の概要

トピック:

- [Force.com REST リソースの理解](#)
- [圧縮の使用](#)
- [REST での cURL の使用例](#)
- [認証について](#)

REST APIでは、Force.com を操作するための強力で便利な使いやすい Web サービス API を提供します。統合および開発が容易という利点があり、モバイルアプリケーションおよび Web 2.0 プロジェクトで使用するための技術では最適な選択です。ただし、処理するレコード件数が多い場合、REST 規則に基づいており、大規模データセットの処理用に最適化されている Bulk API を使用することをお勧めします。

REST API は、SOAP API と同様の基盤データモデルと標準オブジェクトを使用します。詳細は、『[SOAP API 開発者ガイド](#)』を参照してください。REST API も SOAP API と同じ制限に従います。『SOAP API 開発者ガイド』の『[実装に関する考慮事項](#)』セクションを参照してください。

このドキュメントを使用するには、ソフトウェア開発、Web サービス、そして Salesforce ユーザーインターフェースについての基本的な知識が必要です。

このセクションでは、次の内容を理解できます。

- REST API の主要な特性とアーキテクチャ。これにより、アプリケーションにおける Force.com REST リソースの最適な使用方法を理解できます。
- 開発環境の設定方法。設定すると、直ちに REST API を使用できるようになります。
- REST API の使用方法。クイックスタートに従って、ステップごとに一般的な使用事例を学びます。

Force.com REST リソースの理解

REST リソースは、1つのデータレコード、レコードのコレクション、さらには動的なリアルタイム情報などの情報を抽象化したものです。Force.com REST API の各リソースは、名前付きの URI で識別され、標準 HTTP メソッド (HEAD、GET、POST、PATCH、DELETE) を使用してアクセスされます。Force.com REST API は、リソース、その URI、およびそれらの間のリンクの使用に基づきます。リソースは、Salesforce 組織や Force.com 組織との連携に使用します。次のような操作が可能です。

- 使用できる API バージョンに関する概要情報を取得する。
- Account (取引先) などの Salesforce オブジェクトやカスタムオブジェクトに関する詳細情報を取得する。
- User (ユーザ) などの Force.com オブジェクトやカスタムオブジェクトに関する詳細情報を取得する。
- クエリまたは検索を実行する。
- レコードを更新または削除する。

たとえば、Salesforce のバージョンに関する情報を取得するとします。これを行うには、[Versions](#) リソースの要求を送信します (この例では、*na1* インスタンスで cURL を使用します)。

```
curl https://na1.salesforce.com/services/data/
```

この要求の出力は、次のとおりです。

```
[
  {
    "version": "20.0",
    "url": "/services/data/v20.0",
    "label": "Winter '11"
  }
  ...
]
```



メモ: Salesforce は複数のサーバインスタンス上で実行されます。このガイドの例では、*na1* インスタンスが使用されています。組織が使用するインスタンスは異なる場合があります。

Force.com REST API リソースとアーキテクチャの重要な特性を次に示します。

ステートレス

クライアントからサーバへの各要求には、要求を実行するのに必要なすべての情報が含まれている必要があります。サーバに保存されたコンテキストは使用されません。ただし、リソースの表現は URL によって相互接続されるため、ステートが変わってもクライアントは処理を続行できます。

キャッシュの動作

応答にはキャッシュ可能かどうかを示すラベルが付加されます。

統一されたインターフェース

すべてのリソースには、HTTP を介した汎用インターフェースを使用してアクセスします。

名前付きリソース

すべてのリソースは、Force.com URI に続くベース URI を使って名前が付けられます。

階層化されたコンポーネント

Force.com REST API アーキテクチャでは、クライアントとリソースの間にプロキシサーバやゲートウェイなどを介在させることができます。

認証

Force.com REST API は OAuth 2.0 (セキュアな API 認証を可能にするオープンプロトコル) をサポートします。詳細は「[認証について](#)」を参照してください。

JSON および XML のサポート

デフォルトは JSON です。HTTP ACCEPT ヘッダーを使って、JSON と XML のいずれかを選択したり、URI の最後に .json、.xml を付加したりできます (例: /Account/001D000000INjVe.json)。

JavaScript Object Notation (JSON) 形式は UTF-8 でサポートされます。日時情報は ISO8601 形式です。

XML 逐次化は SOAP API と類似しています。XML 要求は UTF-8 および UTF-16 でサポートされ、XML 応答は UTF-8 で提供されます。

圧縮の使用

REST API は、HTTP 1.1 の仕様に定義された標準を使用した要求と応答の圧縮をサポートしています。圧縮は、いくつかのクライアントでは自動的にサポートされており、他のクライアントにも手動で追加できます。クライアント別の詳細は、[Developer Force](#)を参照してください。



ヒント: パフォーマンス向上のため、HTTP 1.1 の仕様に従ったクライアント側での圧縮のサポートをお勧めします。

圧縮を使用するには、要求に HTTP ヘッダー Accept-Encoding: gzip または Accept-Encoding: deflate を含めます。クライアントでこのヘッダーが正しく指定されている場合、REST API は応答を圧縮します。応答には、ヘッダー Content-Encoding: gzip または Content-Encoding: deflate が含まれます。また、Content-Encoding: gzip または Content-Encoding: deflate ヘッダーを含めることによって要求を圧縮することもできます。

応答の圧縮

REST API は、必要に応じて応答を圧縮することができます。応答は、クライアントが Accept-Encoding ヘッダーを送信する場合にのみ圧縮されます。REST API で応答を圧縮する必要がない場合も Accept-Encoding を指定しておけば、指示どおり圧縮を実行します。REST API が応答を圧縮する場合、Content-Encoding ヘッダーも指定します。

要求の圧縮

クライアントは要求を圧縮することもできます。REST API は、処理前にすべての要求を展開します。クライアントは、要求に適切な圧縮アルゴリズムの名前を記した Content-Encoding HTTP ヘッダーを送信する必要があります。詳細は、以下を参照してください。

- Content-Encoding については、www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.11
- Accept-Encoding については、www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.3
- コンテンツのコーディングについては、www.w3.org/Protocols/rfc2616/rfc2616-sec3.html#sec3.5

REST での cURL の使用例

このガイドの例では、Force.com プラットフォームの REST リソースへのアクセス、作成、および操作を行うために、HTTP 要求を送信する cURL ツールを使用します。cURL は、多くの Linux システムや Mac システムにあらかじめインストールされています。Windows バージョンは、curl.haxx.se/ からダウンロードできます。Windows で HTTPS を使用する場合、システムが SSL の cURL 要件を満たしていることを確認してください。



メモ: cURL はオープンソースのツールで、salesforce.com ではサポートされていません。

Mac および Linux システムでのセッション ID のエスケープまたは一重引用符の使用

REST リソースで cURL の例を実行するとき、セッション ID 引数の感嘆符の特殊文字によって、Mac および Linux システムでエラーが発生する場合があります。このエラーの発生を回避するには、次のいずれかを実行します。

- セッション ID が二重引用符で囲まれている場合、セッション ID の感嘆符 (!) 特殊文字の前にバックslash を挿入して (\!) エスケープします。たとえば、この cURL コマンドのセッション ID 文字列では、感嘆符 (!) がエスケープされています。

```
curl https://instance_name.salesforce.com/services/data/v29.0/  
-H "Authorization: Bearer  
00D50000000IehZ\!AQcAQH0dMHZfz972Szmpkb58urFRkgeBGsxL_QJWwYmFAbUeeG7c1E6  
LYUfiDUkWe6H34r1AAwOR8B8fLEz6n04NPGRrq0FM"
```

- セッション ID を一重引用符で囲みます。次に例を示します。

```
curl https://instance_name.salesforce.com/services/data/v29.0/  
-H 'Authorization: Bearer sessionID'
```

認証について

Salesforce では、認証を使用して、ユーザがユーザ名やパスワードのログイン情報を明らかにすることなくセキュアにデータにアクセスできるようにします。

REST API コールを行う前に、OAuth 2.0 を使用してユーザを認証する必要があります。そのためには、次の手順を実行する必要があります。

- Salesforce に [リモートアクセスアプリケーション定義を設定](#)します。
- 使用する正確な [OAuth エンドポイント](#) を決定します。
- 複数の異なる OAuth 2.0 認証フローのいずれかを介してユーザを認証します。OAuth 認証フローには、アプリケーションと Salesforce の間の認証プロセスを調整するために使用する一連の手順が定義されています。次のような OAuth フローがサポートされます。
 - ◇ [Web サーバフロー](#)。サーバがセキュアにコンシューマの秘密を保護できます。
 - ◇ [ユーザエージェントフロー](#)。コンシューマの秘密をセキュアに保存できないアプリケーションによって使用されます。
 - ◇ [ユーザ名パスワードフロー](#)。アプリケーションがユーザログイン情報に直接アクセスします。

ユーザが正常に認証されるとアクセストークンが送られてきます。このアクセストークンを使用して、認証された REST API コールを実行できます。

接続アプリケーションの定義

OAuth を使用して認証するには、Salesforce に接続アプリケーションを定義する必要があります。

リモートアクセスアプリケーションとは、OAuth プロトコルを使用して Salesforce ユーザと外部アプリケーションの両方を検証する、Salesforce の外部アプリケーションです。リモートアクセスアプリケーションは、Salesforce ヘルプの 接続アプリケーションとして実装されています。Salesforce での認証が必要な新規外部アプリケーションを開発する場合、Salesforce にこの新規認証エントリポイントの情報を伝える新規接続アプリケーションを定義する必要があります。

新規接続アプリケーションを作成するには、次の手順を実行します。

1. [設定] から [作成] > [アプリケーション] をクリックし、[新規] をクリックします。
2. アプリケーションの名前を入力します。
3. 取引先責任者のメール情報と、アプリケーションに応じたその他の情報を入力します。
4. [OAuth 設定の有効化] を選択します。
5. [コールバック URL] を入力します。使用する OAuth フローに応じて、これは通常、認証が成功した後にユーザのブラウザがリダイレクトされる URL になります。この URL は一部の OAuth フローでアクセストークンを渡すために使用されるため、URL はセキュア HTTP (HTTPS) またはカスタム URI スキームを使用する必要があります。
6. サポートされているすべての OAuth 範囲を [選択した OAuth 範囲] に追加します。これらの範囲とは、接続アプリケーションを実行するユーザによって付与される権限を示します。
7. [情報 URL] の URL を入力します。ユーザがこの URL にアクセスすると、アプリケーションの詳細を参照できます。
8. [保存] をクリックします。[コンシューマ鍵] が作成され、表示されます。また、[コンシューマの秘密] が作成されます (表示するにはリンクをクリックします)。

リモートアクセスアプリケーションを定義したら、コンシューマ鍵とコンシューマの秘密を使用してアプリケーションを認証します。接続アプリケーションについての詳細は、Salesforce オンラインヘルプを参照してください。

OAuth エンドポイントについて

OAuth エンドポイントとは、Salesforce に対する OAuth 認証要求を行うために使用する URL です。

アプリケーションで認証要求を発行する場合、正確な Salesforce OAuth エンドポイントを使用する必要があります。主要な OAuth エンドポイントは次のとおりです。

- 認証: <https://login.salesforce.com/services/oauth2/authorize>
- トークン要求: <https://login.salesforce.com/services/oauth2/token>
- OAuth トークンの取り消し: <https://login.salesforce.com/services/oauth2/revoke>

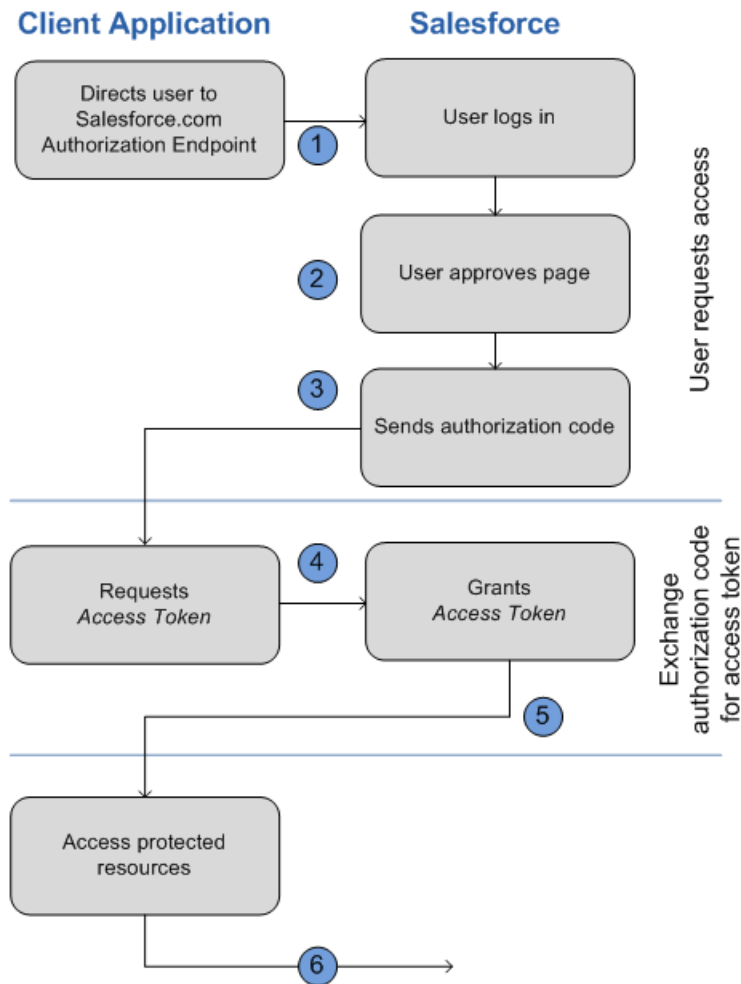
すべてのエンドポイントでセキュア HTTP (HTTPS) が必要です。各 OAuth フローには、使用する必要があるエンドポイントと指定する必要がある要求データが定義されています。

Sandbox 組織で認証を検証する場合、上記に挙げたすべての OAuth エンドポイントで「login.salesforce.com」の代わりに「test.salesforce.com」を使用してください。

Web サーバ OAuth 認証フローについて

安全なサーバでホストされているアプリケーションでは、Web サーバ認証フローを使用します。Web サーバフローにおける重要な点は、サーバがコンシューマの秘密を保護できる必要があります。

このフローでは、クライアントアプリケーションは、ユーザを認証してアプリケーションに認証コードを送信する他の Web サーバまたはリソースにユーザをリダイレクトするように認証サーバに要求します。アプリケーションは認証コードを使用してアクセストークンを要求します。このフローの手順は、次のとおりです。



1. アプリケーションはユーザを適切な Salesforce 認証エンドポイント (<https://login.salesforce.com/services/oauth2/authorize> など) にリダイレクトします。次のパラメータは必須です。

パラメータ	説明
response_type	この認証フローの場合、code にする必要があります。
client_id	リモートアクセスアプリケーション定義の [コンシューマ鍵]。

パラメータ	説明
redirect_url	リモートアクセスアプリケーション定義の [コールバック URL]。

次のパラメータは省略可能です。

パラメータ	説明
display	ログインページの表示の種類を変更します。有効な値は、次のとおりです。 <ul style="list-style-type: none"> page — 全画面のページ認証。これは、値が指定されていない場合のデフォルト値です。 popup — 最新の Web ブラウザのポップアップウィンドウ用に最適化されたコンパクトなダイアログ。 touch — Android や iPhone など、最新のスマートフォン用に設計されたモバイル用に最適化されたダイアログ。 mobile — BlackBerry OS 5 など、タッチスクリーンをサポートしていないスマートフォン用に設計された、モバイル用に最適化されたダイアログ。
immediate	ログインと承認についてユーザにプロンプトメッセージを表示するかどうかを決定します。値は、true か false のいずれかです。デフォルトは false です。 <ul style="list-style-type: none"> true に設定され、ユーザが現在ログインしており、以前にこのアプリケーションを承認している場合、承認ステップはスキップされます。 true に設定され、ユーザがログインしていないか、これまでこのアプリケーションを承認したことがない場合、セッションはただちにエラーコード immediate_unsuccessful で終了します。
state	承認後にコールバック URL で返される、追加の URL 符号化された状態データを指定します。
scope	アプリケーションがアクセスできるデータを指定します。詳細は、オンラインヘルプの「範囲パラメータの値」を参照してください。

認証 URL の例は、次のようになります。

```
https://login.salesforce.com/services/oauth2/authorize?response_type=code&client_id=3MVG9lKcPoNINVBIpJdw1J9LLM82HnFVVX19KYluA5mu0QqEWhqKpoW3svG3XHrXDiCQjKlmdgAvhCscA9GE&redirect_uri=https%3A%2F%2Fwww.mysite.com%2Fcode_callback.jsp&state=mystate
```

2. ユーザが自分のログイン情報で Salesforce にログインします。ユーザは認証エンドポイントを直接操作するため、アプリケーションがユーザのログイン情報を認識することはありません。ログインに成功したら、ユーザはアプリケーションを認証するように要求されます。ユーザがすでにアプリケーションを認証している場合、このステップはスキップされます。
3. クライアントアプリケーションが認証されたことが Salesforce で確認されると、エンドユーザの Web ブラウザは、`redirect_uri` パラメータで指定されたコールバック URL にリダイレクトされます。Salesforce は、認証情報を次の値でリダイレクト URL に付加します。

パラメータ	説明
<code>code</code>	コンシューマがアクセストークンと更新トークンを取得するために使用する必要がある認証コード。
<code>state</code>	最初の要求の一部として渡される状態値 (該当する場合のみ)。

認証情報が付属するコールバック URL の例は、次のようになります。

```
https://www.mysite.com/authcode_callback?code=aWekysIEeqM9PiThEfm0Cnr6MoLIfwYRJcqOqHdF8f9INokharAS09ia7UNP6RiVScerfhc4w%3D%3D
```

4. アプリケーションは認証コードを抽出して、これをアクセストークン要求に含めて Salesforce に渡す必要があります。この要求は、適切な Salesforce トークン要求エンドポイント (<https://login.salesforce.com/services/oauth2/token> など) に対して送信される POST 要求です。次のパラメータは必須です。

パラメータ	説明
<code>grant_type</code>	このフローの値は <code>authorization_code</code> である必要があります。
<code>client_id</code>	リモートアクセスアプリケーション定義の [コンシューマ鍵]。
<code>client_secret</code>	リモートアクセスアプリケーション定義の [コンシューマの秘密]。
<code>redirect_uri</code>	リモートアクセスアプリケーション定義の [コールバック URL]。
<code>code</code>	コンシューマがアクセストークンと更新トークンを取得するために使用する必要がある認証コード。

次のパラメータは省略可能です。

パラメータ	説明
<code>format</code>	期待される戻り形式。デフォルトは <code>json</code> です。値は次のとおりです。 <ul style="list-style-type: none"> <code>urlencoded</code>

パラメータ	説明
	<ul style="list-style-type: none"> • json • xml <p>返される形式は、要求のヘッダーに次のいずれかを使用して指定することもできます。</p> <ul style="list-style-type: none"> • Accept: application/x-www-form-urlencoded • Accept: application/json • Accept: application/xml

アクセストークン POST 要求の例は、次のようになります。

```
POST /services/oauth2/token HTTP/1.1
Host: login.salesforce.com
grant_type=authorization_code&code=aPrxsmIEeqM9PiQroGEWx1UiMQd95_5JUJ
VEhsOFhS8EVvbfYBBJli2W5fn3zbo.8hojaNW_lg%3D%3D&client_id=3MVG9lKcPoNI
NVBIPJjdw1J9LLM82HnFVVX19KYluA5mu0QqEWhqKpoW3svG3XHXDiCQjKlmdgAvhCs
cA9GE&client_secret=1955279925675241571&
redirect_uri=https%3A%2F%2Fwww.mysite.com%2Fcode_callback.jsp
```

5. この要求が成功した場合、サーバは次の内容を持つレスポンスボディを返します。

パラメータ	説明
access_token	アプリケーションが要求を行うために使用するセッション ID として機能するアクセストークン。このトークンは、ユーザログイン情報と同様に保護する必要があります。
refresh_token	新しいアクセストークンを取得するために将来使用できるトークン。  警告: この値は秘密です。ユーザのパスワードなどと同様に処理し、適切な手段で保護する必要があります。
instance_url	API コールの送信先となる Salesforce インスタンスを示します。
id	ユーザ、およびユーザの詳細に関するクエリの両方を識別するために使用できる ID URL。エンドユーザに関する詳細な情報を取得するための HTTP 要求で使用できます。
issued_at	署名が作成された日時。UNIX エポック (1970 年 1 月 1 日 00:00:00 UTC) からの秒数として表されます。
signature	連結 ID と issued_at 値を含むコンシューマの非公開鍵で署名されている Base64 符号化された HMAC-SHA256 署名。この signature は、ID URL

パラメータ	説明
	がサーバから送信された後に変更されていないことを確認するために使用できます。

JSON レスポンスボディの例は、次のようになります。

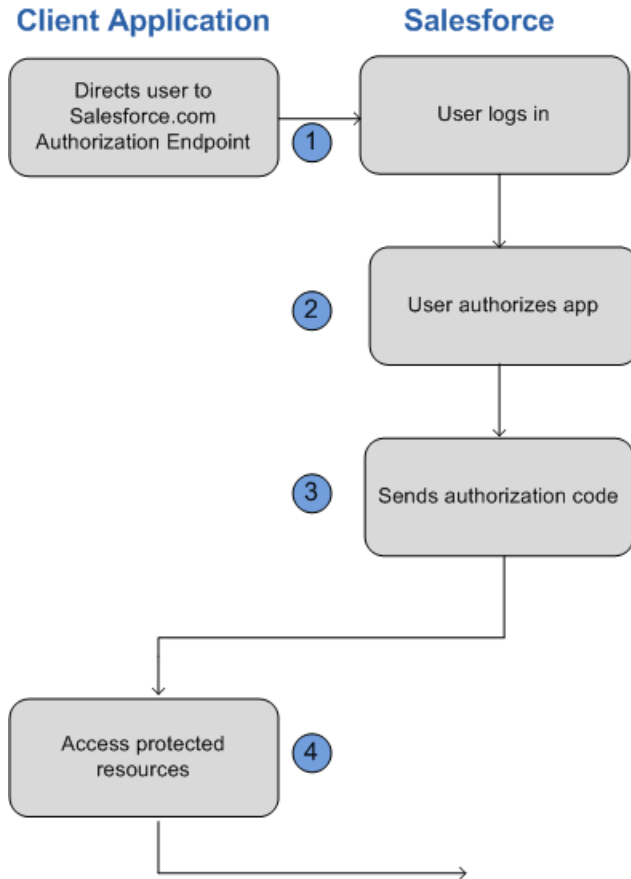
```
{
  "id": "https://login.salesforce.com/id/00Dx0000000BV7z/005x00000012Q9P",
  "issued_at": "1278448101416",
  "refresh_token": "5Aep8614iLM.Dq661ePDmPEgaAW9Oh_L3JKkDpB4xReb54_pZebnUG0h6Sb4KUVDPNtWEofWM39yg==",
  "instance_url": "https://na1.salesforce.com",
  "signature": "CMJ4l+CCaPQiKjoOEwEig9H4wqhpuLSk4J2urAe+fVg=",
  "access_token": "00Dx0000000BV7z!AR8AQP0jITN80ESESj5EbaZTFG0RNBaT1cyWk7TrqoDjoNIWQ2ME_sTZzBjfmOE6zMHq6y8PIW4eWze9JksNEkWU1.Cju7m4"}
}
```

- アプリケーションは、提供されたアクセストークンと更新トークンを使用して保護されたユーザデータにアクセスします。

ユーザエージェント OAuth 認証フローについて

ユーザエージェント認証フローは、ユーザのデバイスにあるクライアントアプリケーション(コンシューマ)で使用されます。これは、JavaScript などのスクリプト言語を使用するブラウザ内で、または携帯機器またはデスクトップアプリケーションから実装することができます。これらのコンシューマは顧客の秘密の機密を保持することができません。

このフローでは、クライアントアプリケーションは、アクセストークンを抽出してアプリケーションに戻ることができる他の Web サーバまたはリソースにユーザをリダイレクトするように認証サーバに要求します。このフローの手順は、次のとおりです。



1. アプリケーションはユーザを適切な Salesforce 認証エンドポイント (<https://login.salesforce.com/services/oauth2/authorize> など) にリダイレクトします。次のパラメータは必須です。

パラメータ	説明
response_type	この認証フローの場合、必ず token にする
client_id	リモートアクセスアプリケーション定義の [コンシューマ鍵]。
redirect_url	リモートアクセスアプリケーション定義の [コールバック URL]。

次のパラメータは省略可能です。


パラメータ	説明
display	<p>ログインページの表示の種類を変更します。有効な値は、次のとおりです。</p> <ul style="list-style-type: none"> • page — 全画面のページ認証。これは、値が指定されていない場合のデフォルト値です。 • popup — 最新の Web ブラウザのポップアップウィンドウ用に最適化されたコンパクトなダイアログ。

パラメータ	説明
	<ul style="list-style-type: none"> touch — Android や iPhone など、最新のスマートフォン用に設計されたモバイル用に最適化されたダイアログ。 mobile — BlackBerry OS 5 など、タッチスクリーンをサポートしていないスマートフォン用に設計された、モバイル用に最適化されたダイアログ。
scope	アプリケーションがアクセスできるデータを指定します。詳細は、オンラインヘルプの「範囲パラメータの値」を参照してください。
state	承認後にコールバック URL で返される、追加の URL 符号化された状態データを指定します。

認証 URL の例は、次のようになります。

```
https://login.salesforce.com/services/oauth2/authorize?response_type=token&
client_id=3MVG91KcPoNINVBIPJjdwlJ9LLJbP_pgwoJYyuisjQhr_LLurNDv7AgQvDTZwCoZuD
ZrXcPCmBv4o.8ds.5iE&redirect_uri=https%3A%2F%2Fwww.mysite.com%2Fuser_callback.jsp&
state=ystate
```

- ユーザが自分のログイン情報で Salesforce にログインします。ユーザは認証エンドポイントを直接操作するため、アプリケーションがユーザのログイン情報を認識することはありません。
- 認証されると、認証エンドポイントはユーザをリダイレクト URL にリダイレクトします。この URL は、アプリケーション用に作成されたリモートアクセスアプリケーションに定義されています。Salesforce は、アクセストークン情報を次の値でリダイレクト URL に付加します。

パラメータ	説明
access_token	アプリケーションが要求を行うために使用するセッション ID として機能するアクセストークン。このトークンは、ユーザログイン情報と同様に保護する必要があります。
expires_in	アクセストークンが有効な時間 (秒)。
refresh_token	<p>新しいアクセストークンを取得するために将来使用できるトークン。</p> <p> 警告: この値は秘密です。ユーザのパスワードなどと同様に処理し、適切な手段で保護する必要があります。</p> <p>更新トークンが返されるのは、リダイレクト URI が <code>https://login.salesforce.com/services/oauth2/success</code> であるか、HTTPS 以外のカスタムプロトコルで使用されている場合のみです。</p>

パラメータ	説明
state	最初の要求の一部として渡される状態値 (該当する場合のみ)。
instance_url	API コールの送信先となる Salesforce インスタンスを示します。
id	ユーザ、およびユーザの詳細に関するクエリの両方を識別するために使用できる ID URL。エンドユーザに関する詳細な情報を取得するための HTTP 要求で使用できます。
issued_at	署名が作成された日時。UNIX エポック (1970 年 1 月 1 日 00:00:00 UTC) からの秒数として表されます。
signature	連結 ID と issued_at 値を含むコンシューマの非公開鍵で署名されている Base64 符号化された HMAC-SHA256 署名。この signature は、ID URL がサーバから送信された後に変更されていないことを確認するために使用できます。

アクセス情報がハッシュ記号 (#) の後に付加されたコールバック URL の例は、次のようになります。

```
https://www.mysite.com/user_callback.jsp#access_token=00Dx0000000BV7z%21AR8
AQBm8J_xr9kLqmZIRyQxZgLCm4HV141aGtW0qW3JCzf5xdTGGGSoVim8FfJkZEqxbjaFbberKGk
8v8AnYrvChG4qJbQo8&refresh_token=5Aep8614iLM.Dq661ePDmPEgaAW9Oh_L3JKkDpB4xR
eb54_pZfVtildPEk8aimw4Hr9ne7VXXVSIQ%3D%3D&expires_in=7200&state=mystate
```

- アプリケーションは、提供されたアクセストークンと更新トークンを使用して保護されたユーザデータにアクセスします。

ユーザエージェント OAuth フローを使用するときの考慮事項は、次のとおりです。

- アクセストークンは符号化され、リダイレクト URI になっているため、エンドユーザや、コンピュータまたはデバイス上にある他のアプリケーションに公開できます。JavaScript を使用して認証する場合、`window.location.replace()` をコールし、ブラウザの履歴からコールバックを削除します。

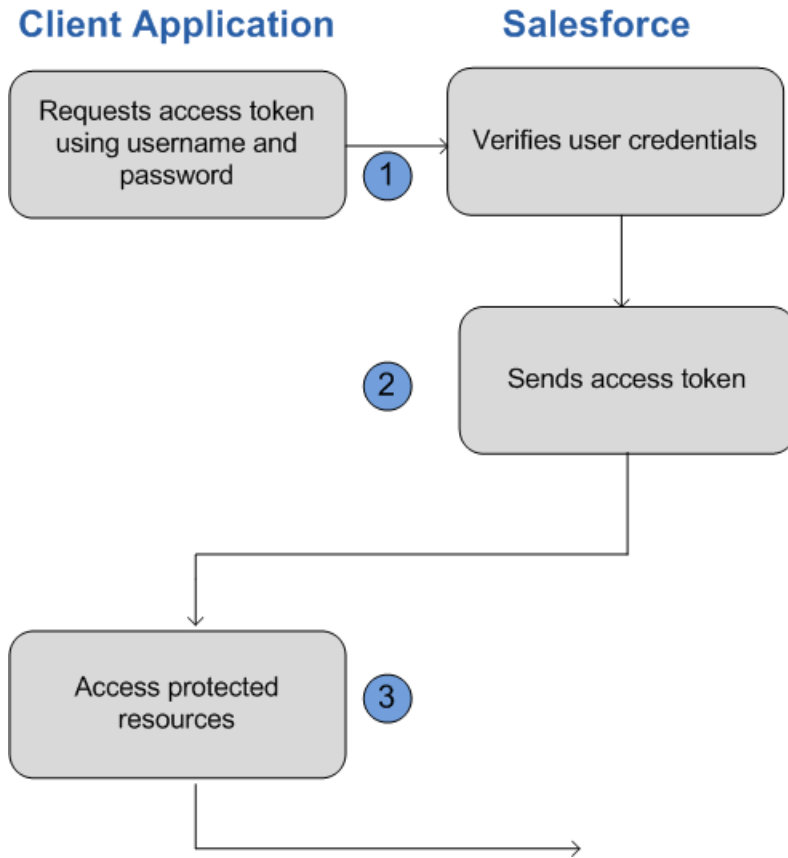
ユーザ名パスワード OAuth 認証フローについて

コンシューマにすでにユーザの認証情報がある場合、ユーザ名パスワード認証フローを使用して認証できます。


このフローでは、次の手順のようにアプリケーションがユーザのログイン情報を使用してアクセストークンを要求します。



警告: この OAuth 認証フローでは、ユーザの信頼情報をやり取りする必要があります。この認証フローは、必要な場合にのみ使用してください。更新トークンは発行されません。



1. コンシューマはユーザのユーザ名とパスワードを使用してアクセストークンを要求します。これを行うには、適切な Salesforce トークン要求エンドポイント (<https://login.salesforce.com/services/oauth2/token> など) に対して帯域外 POST 要求を行います。次の要求項目は必須です。

パラメータ	説明
grant_type	この認証フローの場合、password にする必要があります。
client_id	リモートアクセスアプリケーション定義の [コンシューマ鍵]。
client_secret	リモートアクセスアプリケーション定義の [コンシューマの秘密]。
username	エンドユーザのユーザ名。
password	エンドユーザのパスワード。
 メモ: ユーザのセキュリティトークンをユーザのパスワードに付加する必要があります。セキュリティトークンは、Salesforce で自動生成されたキーです。たとえば、ユーザのパスワードが mypassword で、セキュリティトークンが XXXXXXXXXXXX の場合は、このパラメータには、値 mypasswordXXXXXXXXXXXX を指定	

パラメータ	説明
	する必要があります。セキュリティトークンの詳細は、オンラインヘルプの「セキュリティトークンのリセット」を参照してください。

リクエストボディの例は、次のようになります。

```
grant_type=password&client_id=3MVG9lKcPoNINVBIPJjdwlJ9LLM82Hn
FVVX19KYluA5mu0QqEWhqKpoW3svG3XhrXDICQjKlmdgAvhCscA9GE&client_secret=
1955279925675241571&username=testuser%40salesforce.com&password=mypassword123456
```

2. Salesforce はユーザログイン情報を検証し、成功したら、応答とアクセストークンをアプリケーションに送信します。この応答には次の値が含まれます。

パラメータ	説明
access_token	アプリケーションが要求を行うために使用するセッション ID として機能するアクセストークン。このトークンは、ユーザログイン情報と同様に保護する必要があります。
instance_url	API コールの送信先となる Salesforce インスタンスを示します。
id	ユーザ、およびユーザの詳細に関するクエリの両方を識別するために使用できる ID URL。エンドユーザに関する詳細な情報を取得するための HTTP 要求で使用できます。
issued_at	署名が作成された日時。UNIX エポック (1970 年 1 月 1 日 00:00:00 UTC) からの秒数として表されます。
signature	連結 ID と issued_at 値を含むコンシューマの非公開鍵で署名されている Base64 符号化された HMAC-SHA256 署名。この signature は、ID URL がサーバから送信された後に変更されていないことを確認するために使用できます。

レスポンスボディの例は、次のようになります。

```
{"id":"https://login.salesforce.com/id/00Dx0000000BV7z/005x00000012Q9P",
"issued_at":"1278448832702","instance_url":"https://na1.salesforce.com",
"signature":"0CmxinZir53Yex7nE0TD+zMpIWIYGb/bdJh6XfOH6EQ=", "access_token":
"00Dx0000000BV7z!AR8AQAxo9UfVkh8AlV0Gomt9Czx9LjHnSSpwBMmbRcgKFmxOtvxjTrKWl
9ye6PE3Ds1eQz3z8j3W7_VbWmEu4Q8TVGSTHxs"}
```

3. アプリケーションは、提供されたアクセストークンを使用して保護されたユーザデータにアクセスします。ユーザエージェント OAuth フローを使用するときの考慮事項は、次のとおりです。

- このフローではユーザが Salesforce でログインするためにリダイレクトされることはないため、ユーザは直接アプリケーションを認証できません。そのため、更新トークンは使用できません。アプリケーションで更新トークンが必要な場合、Web サーバまたはユーザエージェント OAuth フローの使用を検討してください。

OAuth 更新トークンプロセスについて

Web サーバ OAuth 認証フローとユーザエージェントフローはどちらも、新しいアクセストークンの取得に使用可能な更新トークンを提供します。

アクセストークンは、Salesforce のセッションタイムアウトで指定された有効期間に制限されています。アプリケーションが有効期限の切れたアクセストークンを使用すると、「Session expired or invalid」エラーが返されます。アプリケーションが Web サーバまたはユーザエージェント OAuth 認証フローを使用している場合、認証中に更新トークンが提供され、新しいアクセストークンの取得に使用できる可能性があります。

クライアントアプリケーションが新しいアクセストークンを取得するには、次の要求パラメータを指定して POST 要求をトークン要求エンドポイントに送信します。

パラメータ	説明
grant_type	値は refresh_token である必要があります。
refresh_token	クライアントアプリケーションがすでに受け取っている更新トークン。
client_id	リモートアクセスアプリケーション定義の [コンシューマ鍵]。
client_secret	リモートアクセスアプリケーション定義の [コンシューマの秘密]。このパラメータは省略可能です。
format	<p>期待される戻り形式。デフォルトは、json です。値は次のとおりです。</p> <ul style="list-style-type: none"> urlencoded json xml <p>返される形式は、要求のヘッダーに次のいずれかを使用して指定することもできます。</p> <ul style="list-style-type: none"> Accept: application/x-www-form-urlencoded Accept: application/json Accept: application/xml <p>このパラメータは省略可能です。</p>

更新トークン POST 要求の例は、次のようになります。

```
POST /services/oauth2/token HTTP/1.1
Host: https://login.salesforce.com/
grant_type=refresh_token&client_id=3MVG9lKcPoNINVBIPJjdwlJ9LLM82HnFVVX19KYluA5mu0
QqEWhqKpoW3svG3XHrXDiCQjKlmdgAvhCscA9GE&client_secret=1955279925675241571
&refresh_token=your token here
```

Salesforce は、更新トークン要求を検証した後、次のレスポンスボディパラメータを使用してアプリケーションに応答を送信します。

パラメータ	説明
access_token	アプリケーションが要求を行うために使用するセッション ID として機能するアクセストークン。このトークンは、ユーザログイン情報と同様に保護する必要があります。
instance_url	API コールの送信先となる Salesforce インスタンスを示します。
id	ユーザ、およびユーザの詳細に関するクエリの両方を識別するために使用できる ID URL。エンドユーザに関する詳細な情報を取得するための HTTP 要求で使用できます。
issued_at	署名が作成された日時。UNIX エポック (1970 年 1 月 1 日 00:00:00 UTC) からの秒数として表されます。
signature	連結 ID と issued_at 値を含むコンシューマの非公開鍵で署名されている Base64 符号化された HMAC-SHA256 署名。この signature は、ID URL がサーバから送信された後に変更されていないことを確認するために使用できます。

JSON レスポンスボディの例は、次のようになります。

```
{ "id": "https://login.salesforce.com/id/00Dx0000000BV7z/005x00000012Q9P",
  "issued_at": "1278448384422", "instance_url": "https://na1.salesforce.com",
  "signature": "SSSbLO/gBhmmyNUvN180DBDFYHzakxOMgqYtu+hDPsc=",
  "access_token": "00Dx0000000BV7z!AR8AQP0jITN80ESEsj5EbaZTFG0RNBaT1cyWk7TrqoDjoNIWQ2ME_sTZzBjfmOE6zMHq6y8PIW4eWze9JksNEkWU1.Cju7m4" }
```

更新トークン OAuth プロセスを使用するときの考慮事項は、次のとおりです。

- アクセストークンのセッションタイムアウトを Salesforce で設定するには、[設定] から [セキュリティのコントロール] > [セッションの設定] をクリックします。
- アプリケーションがユーザ名パスワード OAuth 認証フローを使用する場合、このフローではユーザはアプリケーションを認証できないため、更新トークンは発行されません。アクセストークンの有効期限が切れた場合、ユーザ名パスワード OAuth フローを使用するアプリケーションはユーザを再認証する必要があります。

その他のリソースを見つける

次のリソースは、Salesforce で OAuth を使用するときの関連情報を説明しています。

- [Digging Deeper into OAuth on Force.com \(Force.com の OAuth の詳細\)](#)
- [Using OAuth to Authorize External Applications \(OAuth を使用した外部アプリケーションの認証\)](#)

OAuthを実装するサードパーティクライアントライブラリとして、次のようなリソースを必要に応じて参考にしてください。

- Ruby on Rails: [OmniAuth](#)
- Java: [Apache Amber](#)
- その他の OAuth クライアントライブラリ: [OAuth.net](#)

第2章

クイックスタート

トピック:

- [前提条件](#)
- [ステップ 1: Salesforce Developer Edition 組織を取得する](#)
- [ステップ 2: 認証を設定する](#)
- [ステップ 3: cURL で HTTP 要求を送信する](#)
- [ステップ 4: サンプルコードを実行する](#)

開発環境に REST サンプルアプリケーションを作成し、REST API の能力と柔軟性を確認します。

前提条件

次の前提条件をすべて満たすことで、クイックスタートのサンプルの作成と使用が容易になります。

- 製品ドキュメントに従って、開発プラットフォームをインストールする。
- このクイックスタートで REST 要求の実行に使用するツールである cURL に習熟しておく。他のツールを使用する場合は、コード例を変換できるようにそのツールに十分慣れておく必要があります。
- このクイックスタートで使用する JavaScript Object Notation (JSON) に習熟しておく。JSON を使用しない場合は、サンプルを JSON から読み換えられる程度に自分が使用する標準言語に習熟している必要があります。
- アプリケーションサーバの SSL エンドポイントを有効にする。
- OAuth 2.0 に習熟しておく。OAuth 2.0 にはいくつかの設定が必要です。手順は説明しますが、基本的な概念とワークフローを理解しておくに役に立ちます。
- このクイックスタートを開始する前に、すべての手順に目を通す。このガイドの他の部分を確認しておく、用語や概念を把握できます。

ステップ 1: Salesforce Developer Edition 組織を取得する

まだ Force.com 開発者コミュニティのメンバーでない場合、<http://developer.force.com/join> にアクセスし、Developer Edition 組織のサインアップの説明に従ってください。すでに Enterprise Edition、Unlimited Edition、または Performance Edition を所有している場合でも、組織の使用中的数据を保護するために、サンプルデータに対するソリューションの開発、ステージングおよびテストには Developer Edition を使用します。これは、特に、(データをただ参照するだけのアプリケーションに対し)データを挿入、更新または削除するアプリケーションの場合に該当します。

Developer Edition 組織をすでに所有している場合は、「API の有効化」権限があることを確認します。この権限はデフォルトで有効になっていますが、管理者によって変更されている場合があります。詳細は、Salesforce ユーザーインターフェースのヘルプを参照してください。

ステップ 2: 認証を設定する

OAuth 2.0 を使用して、またはセッション ID を渡すことによって、認証を設定できます。



重要: 他のユーザのパスワードを処理している場合は、セッション ID は使用しないでください。

認証用の OAuth コンシューマ ID の取得を希望するパートナー様は salesforce.com まで連絡してください。

OAuth 2.0 の設定

OAuth 2.0 の設定には、Salesforce 内と他の場所でのいくつかのステップを実行する必要があります。ステップに不明な点がある場合は、「[認証について](#)」または Salesforce オンラインヘルプを参照してください。次の例では、[Web サーバ OAuth フロー](#)を使用します。

1. まだ新しいリモートアクセスアプリケーションを作成していない場合は、Salesforce で、[設定] から [開発] > [リモートアクセス] をクリックし、[新規] をクリックして新しいリモートアクセスアプリケーションを作成します。ここで指定する [コールバック URL] は、Web アプリケーションのコールバック URL と同じです。Java を使用する場合、通常はこれはサブレットです。コールバック URL はセキュアである必要があります。

す。http:// は機能せず、https:// のみが機能します。開発環境では、コールバック URL は https://localhost:8443/RestTest/oauth/_callback のような形になります。[保存]をクリックすると、[コンシューマ鍵] が作成されて表示され、[コンシューマの秘密] が作成されます (リンクをクリックして表示します)。



メモ: OAuth 2.0 仕様では、「consumer (コンシューマ)」ではなく「client (クライアント)」という表現が使われます。Salesforce は OAuth 2.0 をサポートします。

次に、Salesforce の項目と、これ以降の手順に出てくるサンプルコード内の項目の対応関係を示します。

- client_id は [コンシューマ鍵] に対応
- client_secret は [コンシューマの秘密] に対応
- redirect_uri は [コールバック URL] に対応

クライアントアプリケーションでは、ユーザを適切な Salesforce 認証エンドポイントにリダイレクトします。ユーザのログインが成功すると、Salesforce は認証コードでリダイレクト URI をコールします。次のステップでは、認証コードを使用してアクセストークンを取得します。

2. Java またはその他のクライアントアプリケーションから、適切な Salesforce トークン要求エンドポイントに要求を行い、grant_type、client_id、client_secret、および redirect_uri を渡します。redirect_uri は、Salesforce がコールバック送信する先の URI です。

```
initParams = {
    @WebInitParam(name = "clientId", value =
        "3MVG9lKcPoNINVBJSOQsNCD.HHDdbugPsNXwwyFbgb47Kwa_PTv"),
    @WebInitParam(name = "clientSecret", value = "5678471853609579508"),
    @WebInitParam(name = "redirectUri", value =
        "https://localhost:8443/RestTest/oauth/_callback"),
    @WebInitParam(name = "environment", value =
        "https://na1.salesforce.com/services/oauth2/token") }

HttpClient httpClient = new HttpClient();
PostMethod post = new PostMethod(environment);
post.addParameter("code", code);
post.addParameter("grant_type", "authorization_code");

/** For session ID instead of OAuth 2.0, use "grant_type", "password" */
post.addParameter("client_id", clientId);
post.addParameter("client_secret", clientSecret);
post.addParameter("redirect_uri", redirectUri);
```

client_id (または consumer key) の値と client_secret (または consumer secret) が有効な場合、Salesforce はコールバックを、access_token. の値が含まれる redirect_uri に指定された URI に送信します。

3. アクセストークンの値を Cookie として保存し、以降のすべての要求で使用します。次に例を示します。

```
//exception handling removed for brevity...
//this is the post from step 2
httpClient.executeMethod(post);
String responseBody = post.getResponseAsString();

String accessToken = null;
JSONObject json = null;
try {
    json = new JSONObject(responseBody);
    accessToken = json.getString("access_token");
    issuedAt = json.getString("issued_at");
    /** Use this to validate session
     * instead of expiring on browser close.
     */
```

```

    } catch (JSONException e) {
        e.printStackTrace();
    }

    HttpServletResponse httpResponse = (HttpServletResponse)response;
    Cookie session = new Cookie(ACCESS_TOKEN, accessToken);
    session.setMaxAge(-1); //cookie not persistent, destroyed on browser exit
    httpResponse.addCookie(session);

```

これで、認証は完了です。

4. 認証されると、各要求は、ヘッダーに `access_token` 値を渡す必要があります。要求パラメータとして渡すことはできません。

```

HttpClient httpClient = new HttpClient();
GetMethod gm = new GetMethod(serviceUrl);

//set the token in the header
gm.setRequestHeader("Authorization", "Bearer "+accessToken);
//set the SOQL as a query param
NameValuePair[] params = new NameValuePair[1];

/**
 * other option instead of query string, pass just the fields you want back:
 * https://instance_name.salesforce.com/services/data/v20.0/subjects/Account/
 * 001D000000INjVe?fields=AccountNumber,BillingPostalCode
 */
params[0] = new NameValuePair("q","SELECT name, title FROM Contact LIMIT 100");
gm.setQueryString(params);

httpClient.executeMethod(gm);
String responseBody = gm.getResponseBodyAsString();
//exception handling removed for brevity
JSONObject json = new JSONObject(responseBody);

JSONArray results = json.getJSONArray("records");

for(int i = 0; i < results.length(); i++)
    response.getWriter().write(results.getJSONObject(i).getString("Name")+      ",
    "+results.getJSONObject(i).getString("Title")+"\n");

```

REST 要求にアクセストークンを提供する構文は、次のとおりです。

```
Authorization: Bearer access_token
```

次に例を示します。

```
curl https://instance_name.salesforce.com/services/data/v20.0/ -H 'Authorization: Bearer
access_token'
```

セッション ID の認証

他のユーザのパスワードを処理していない場合は、OAuth 2.0 アクセストークンの代わりにセッション ID を使用できます。

1. セッション ID を取得します。たとえば、SOAP API `login()` コールは、セッション ID を返します。また、たとえば Apex の現在のコンテキストの一部として、セッション ID を持つこともできます。開発中にテスト

目的のためだけにセッション ID が必要な場合、cURL コマンドで次のようにユーザ名パスワード OAuth フローを使用できます。

```
curl https://login.salesforce.com/services/oauth2/token -d "grant_type=password" -d "client_id=myclientid" -d "client_secret=myclientsecret" -d "mylogin@salesforce.com" -d "password=mypassword123456"
```

クライアント ID、クライアントの秘密、ユーザ名、およびユーザセキュリティトークンを付加したパスワードが必要です。

2. リソースに要求を送信するときに、セッション ID を使用します。ID を *token* 値と置き換えます。構文は同じです。

```
Authorization: Bearer access_token
```

次に例を示します。

```
curl https://instance_name.salesforce.com/services/data/v20.0/ -H 'Authorization: Bearer access_token'
```

ステップ 3: cURL で HTTP 要求を送信する

Force.com REST API を操作するには、HTTP 要求を構築するようにクライアントアプリケーションを設定する必要があります (cURL を使用します)。

クライアントアプリケーションの設定

REST API は、HTTP GET および HTTP POST メソッドを使用して、JSON と XML コンテンツを送信および取得するため、好みのツールまたは言語を使って簡単にクライアントアプリケーションを構築できます。HTTP 要求と応答の送受信を単純化するために、cURL というコマンドラインツールを使用します。

cURL は、多くの Linux システムや Mac システムにあらかじめインストールされています。Windows バージョンは、curl.haxx.se/ からダウンロードできます。Windows で HTTPS を使用する場合、システムが SSL の cURL 要件を満たしていることを確認してください。

REST API リソースを使った HTTP 要求の送信

REST API リソースへの HTTP 要求には、次の情報が含まれている必要があります。

- HTTP メソッド (HEAD、GET、POST、PATCH、または DELETE)。
- 要求の認証に使用される OAuth 2.0 アクセストークン。トークンの取得方法については、「[クイックスタート](#)」 (ページ 19) を参照してください。
- リソースの形式 (XML または JSON)、または URI の `.json` または `.xml` 拡張子を示すために使用する HTTP ACCEPT ヘッダー。デフォルトは JSON です。
- Force.com REST リソース。
- 新しい情報でレコードを更新するなどの要求に必要な情報が含まれる JSON ファイルまたは XML ファイル。

HTTP メソッドは、情報の取得や、レコードの作成、更新、削除など、目的のアクションを示すために使用されます。

- HEAD は、リソースメタデータの取得に使用されます。
- GET は、リソースに関する基本的なサマリー情報など、情報の取得に使用されます。

- POST は、新しいオブジェクトを作成するために使用されます。
- PATCH は、レコードを更新するために使用されます。
- DELETE は、レコードを削除するために使用されます。

リソースにアクセスするには、ヘッダー、メソッド、リソース名を含む HTTP 要求を送信します。

たとえば、newaccount.json という JSON 形式のファイルを使用して、Account レコードを作成するとします。これには、新しい Account レコードに保存される情報が含まれます。

```
{
  "Name" : "test"
}
```

na1 に cURL を使用すると、要求は、次のようになります。

```
curl https://na1.salesforce.com/services/data/v20.0/subjects/Account/ -H "Authorization: Bearer token" -H "Content-Type: application/json" -d @newaccount.json"
```

HTTP 要求ヘッダーは次のようになります。

```
POST /services/data/v20.0/subjects/Account HTTP/1.1
User-Agent: curl/7.19.7 (universal-apple-darwin10.0) libcurl/7.19.7 OpenSSL/0.9.8l zlib/1.2.3
Host: na1.salesforce.com
Accept: */*
Content-Length: 1411
Content-Type: application/json
Authorization: Bearer XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
X-PrettyPrint:1
```

応答は次のようになります。

```
Date: Thu, 21 Oct 2010 22:16:22 GMT
Content-Length: 71
Location: /services/data/v20.0/subjects/Account/001T000000NU96UIAT
Content-Type: application/json; charset=UTF-8 Server:
{ "id" : "001T000000NU96UIAT",
  "errors" : [ ],
  "success" : true }
```

リソースおよび対応する URI のリストについては、「[リファレンス](#)」(ページ 59)を参照してください。

関連リンク

[REST での cURL の使用例](#)

ステップ 4: サンプルコードを実行する

このセクションでは、一連の REST 要求を作成します。要求の構築には cURL を使用し、すべての要求と応答の形式として JSON を使用します。各要求では、REST リソースと共にベース URI を使用します。これらの例のベース URI は <https://na1.salesforce.com/services/data> です。詳細は、「[Force.com REST リソースの理解](#)」(ページ 2)を参照してください。

この例では、一連の REST 要求が次のシナリオで使用されます。

1. [Salesforce バージョン](#)を取得する。
2. [Salesforce バージョン](#)を使用して、使用可能なリソースのリストを取得する。

3. リソースの 1 つを使用して、使用可能なオブジェクトのリストを取得する。
4. オブジェクトの 1 つを選択して、そのメタデータの説明を取得する。
5. その同じオブジェクトの項目のリストを取得する。
6. SOQL クエリを実行して、Account レコードのすべての name 項目の値を取得する。
7. Account レコードの 1 つで請求先市区郡の情報 (BillingCity) を更新する。

Salesforce バージョンを取得する

はじめに、使用可能な各 Salesforce バージョンに関する情報を取得します。これを行うには、[Versions](#) リソースに要求を送信します。この場合、要求に認証は必要ありません。

```
curl https://na1.salesforce.com/services/data/
```

応答ヘッダーを含む、この要求の出力は次のとおりです。

```
Content-Length: 88
Content-Type: application/json;
charset=UTF-8 Server:
[
  {
    "version": "20.0",
    "url": "/services/data/v20.0",
    "label": "Winter '11"
  }
  ...
]
```

出力は、すべての有効なバージョンで使用可能なリソースを指定します (結果には複数の値が含まれる場合があります)。次に、これらのバージョンの 1 つを使用して、それに含まれるリソースを検出します。

リソースのリストを取得する

次のステップは、Salesforce (この例ではバージョン 20.0) で使用可能なリソースのリストを取得します。これを行うには、[Resources by Version](#) の要求を送信します。

```
curl https://na1.salesforce.com/services/data/v20.0/ -H "Authorization: Bearer access_token"
-H "X-PrettyPrint:1"
```

この要求の出力は、次のとおりです。

```
{
  "subjects" : "/services/data/v20.0/subjects",
  "search" : "/services/data/v20.0/search",
  "query" : "/services/data/v20.0/query",
  "recent" : "/services/data/v20.0/recent"
}
```

この出力から、`subjects` が、Salesforce バージョン 20.0 で使用可能なリソースの 1 つであることがわかります。このリソースを次の要求で使用し、使用可能なオブジェクトを取得することができます。

使用可能なオブジェクトのリストを取得する

使用可能なリソースのリストを取得したので、使用可能なオブジェクトのリストを要求できます。これを行うには、[Describe Global](#) の要求を送信します。

```
curl https://na1.salesforce.com/services/data/v20.0/subjects/ -H "Authorization: Bearer access_token" -H "X-PrettyPrint:1"
```

この要求の出力は、次のとおりです。

```
Transfer-Encoding: chunked
Content-Type: application/json;
charset=UTF-8 Server:
{
  "encoding" : "UTF-8",
  "maxBatchSize" : 200,
  "subjects" : [ {
    "name" : "Account",
    "label" : "Account",
    "custom" : false,
    "keyPrefix" : "001",
    "updateable" : true,
    "searchable" : true,
    "labelPlural" : "Accounts",
    "layoutable" : true,
    "activateable" : false,
    "urls" : { "subject" : "/services/data/v20.0/subjects/Account",
    "describe" : "/services/data/v20.0/subjects/Account/describe",
    "rowTemplate" : "/services/data/v20.0/subjects/Account/{ID}" },
    "createable" : true,
    "customSetting" : false,
    "deletable" : true,
    "deprecatedAndHidden" : false,
    "feedEnabled" : false,
    "mergeable" : true,
    "queryable" : true,
    "replicateable" : true,
    "retrieveable" : true,
    "undeletable" : true,
    "triggerable" : true },
  },
  ...
}
```

この出力から、Account オブジェクトが使用可能であることがわかります。Account オブジェクトに関する詳細情報の取得については、次のステップで説明します。

オブジェクトの基本情報を取得する

Account オブジェクトを使用可能なリソースとして特定したため、そのメタデータに関するいくつかの基本情報を取得できます。これを行うには、[sObject Basic Information](#) の要求を送信します。

```
curl https://na1.salesforce.com/services/data/v20.0/subjects/Account/ -H "Authorization: Bearer access_token" -H "X-PrettyPrint:1"
```

この要求の出力は、次のとおりです。

```
{
  "objectDescribe" :
  {
    "name" : "Account",
    "updateable" : true,
    "label" : "Account",
    "keyPrefix" : "001",
  }
}
```



```

...
    "replicateable" : true,
    "retrieveable" : true,
    "undeletable" : true,
    "triggerable" : true
  },
  "recentItems" :
  [
    {
      "attributes" :
      {
        "type" : "Account",
        "url" : "/services/data/v20.0/subjects/Account/001D000000INjVeIAL"
      },
      "Id" : "001D000000INjVeIAL",
      "Name" : "asdasdasd"
    },
    ...
  ]
}

```

この出力から、名前や表示ラベルなど、Account オブジェクトのいくつかの基本属性を確認できます。最近使用された Account レコードのリストも取得されています。長さやデフォルト値など、その項目に関する詳細情報が必要な場合があるため、次のステップで、Account オブジェクトに関する詳細情報の取得について説明します。

項目のリストを取得する

Account オブジェクトのメタデータに関するいくつかの基本情報を取得したので、詳細情報を取得できます。これを行うには、[sObject Describe](#) の要求を送信します。

```

curl https://na1.salesforce.com/services/data/v20.0/subjects/Account/describe/ -H
"Authorization: Bearer access_token" -H "X-PrettyPrint:1"

```

この要求の出力は、次のとおりです。

```

{
  "name" : "Account",
  "fields" :
  [
    {
      "length" : 18,
      "name" : "Id",
      "type" : "id",
      "defaultValue" : { "value" : null },
      "updateable" : false,
      "label" : "Account ID",
      ...
    },
    ...
  ],
  "updateable" : true,
  "label" : "Account",
  ...
  "urls" :
  {
    "uiEditTemplate" : "https://na1.salesforce.com/{ID}/e",
    "subject" : "/services/data/v20.0/subjects/Account",
    "uiDetailTemplate" : "https://na1.soma.salesforce.com/{ID}",
    "describe" : "/services/data/v20.0/subjects/Account/describe",
    "rowTemplate" : "/services/data/v20.0/subjects/Account/{ID}",
    "uiNewRecord" : "https://na1.salesforce.com/001/e"
  }
}

```

```

    },
    "childRelationships" :
    [
        {
            "field" : "ParentId",
            "deprecatedAndHidden" : false,
            ...
        },
        ...
    ],
    "createable" : true,
    "customSetting" : false,
    ...
}

```

この出力から、その項目の属性や子リレーションなど、Account オブジェクトのさらに詳細な情報を確認できます。これで、組織の Account オブジェクトに対する便利なクエリと更新を作成するのに必要な情報を取得できました。次のステップでは、実際にクエリと更新を実行します。

SOQL クエリを実行する

Account オブジェクトの項目名がわかったので、SOQL クエリを実行できます。例として、今回は、すべての取引先名の値のリストを取得します。これを行うには、[Query](#) 要求を送信します。

```

curl https://na1.salesforce.com/services/data/v20.0/query?q=SELECT+name+from+Account -H
"Authorization: Bearer access_token" -H "X-PrettyPrint:1"

```

この要求の出力は、次のとおりです。

```

{
  "done" : true,
  "totalSize" : 14,
  "records" :
  [
    {
      "attributes" :
      {
        "type" : "Account",
        "url" : "/services/data/v20.0/subjects/Account/001D000000IRFmaIAH"
      },
      "Name" : "Test 1"
    },
    {
      "attributes" :
      {
        "type" : "Account",
        "url" : "/services/data/v20.0/subjects/Account/001D000000IomazIAB"
      },
      "Name" : "Test 2"
    },
    ...
  ]
}

```

この出力から使用可能な取引先名のリストを取得します。それぞれの名前のある属性には取引先の ID が含まれます。次のステップでは、この情報を使用して、取引先の 1 つを更新します。



メモ: SOQL についての詳細は、[Salesforce SOQL and SOSL Reference Guide](#)を参照してください。

レコードの項目を更新する

取引先名と ID を把握できたので、レコードの 1 つを取得し、請求先市区郡の情報 (BillingCity) を更新できます。これを行うには、[sObject Rows](#) 要求を送信する必要があります。オブジェクトを更新するには、市区郡に関する新しい情報を提供します。次の情報を含む、patchaccount.json というテキストファイルを作成します。

```
{
  "BillingCity" : "Fremont"
}
```

REST 要求に、この JSON ファイルを指定します。cURL 表記には、データを指定する場合、-d オプションが必要です。詳細は、<http://curl.haxx.se/docs/manpage.html> を参照してください。

また、REST リソースを更新するために使用される PATCH メソッドを指定します。次の cURL コマンドは、ID 項目を使用して指定された Account レコードを取得し、その市区郡を更新します。

```
curl https://na1.salesforce.com/services/data/v20.0/subjects/Account/001D000000IroHJ -H
"Authorization: Bearer access_token" -H "X-PrettyPrint:1" -H "Content-Type: application/json"
--data-binary @patchaccount.json -X PATCH
```

レスポンスボディはなく、ヘッダーのみが返されます。

```
HTTP/1.1 204 No Content
Server:
Content-Length: 0
```

その Account レコードのページを更新すると、請求先住所の市区郡が「Fremont」に変更されていることを確認できます。

その他のリソース

- developer.force.com で「Ruby」を検索してください。
- Ruby での使用開始は、『[Force.com Cookbook](#)』のレシピを参照してください。
- [Force.com REST API のディスカッションボード](#)

REST リソースの使用

第 3 章

REST API リソースの使用

トピック:

- [組織に関する情報の取得](#)
- [オブジェクトメタデータの使用](#)
- [レコードの操作](#)
- [検索とクエリの使用](#)
- [最近参照した情報の操作](#)
- [ユーザパスワードの管理](#)

このセクションでは、例を示しながら、REST API リソースを使用してオブジェクト、組織情報、クエリの使用などさまざまなタスクを実行する方法を説明します。

REST API リソースの完全なリファレンス情報は、「[リファレンス](#)」(ページ 59)を参照してください。

組織に関する情報の取得

REST API を使用して、組織に関する情報を取得できます。

このセクションの例では、REST API リソースを使用して、組織で利用できるすべてのオブジェクトのリストなど、組織レベルの情報を取得します。

使用可能な REST API バージョンをリストする

バージョン、表示ラベル、および各バージョンのルートへのリンクなど、現在使用できる各 REST API バージョンについての概要情報をリストするには、[Versions](#) リソースを使用します。バージョンのリストを取得するための認証は必要ありません。

使用例

```
curl http://na1.salesforce.com/services/data/
```

リクエストボディの例

不要

JSON レスポンスボディの例

```
[
  {
    "version" : "20.0",
    "label" : "Winter '11",
    "url" : "/services/data/v20.0"
  },
  {
    "version" : "21.0",
    "label" : "Spring '11",
    "url" : "/services/data/v21.0"
  },
  ...
  {
    "version" : "26.0",
    "label" : "Winter '13",
    "url" : "/services/data/v26.0"
  }
]
```

XML レスポンスボディの例

```
<?xml version="1.0" encoding="UTF-8"?>
<Versions>
  <Version>
    <label>Winter &apos;11</label>
    <url>/services/data/v20.0</url>
    <version>20.0</version>
  </Version>
  <Version>
    <label>Spring &apos;11</label>
    <url>/services/data/v21.0</url>
    <version>21.0</version>
  </Version>

```

```

...
<Version>
  <label>Winter &apos;13</label>
  <url>/services/data/v26.0</url>
  <version>26.0</version>
</Version>
</Versions>

```

使用可能な REST リソースをリストする

指定された API バージョンで使用可能なリソースをリストするには、[Resources by Version](#) リソースを使用します。このリソースは、その他のリソースそれぞれの名前と URI を提供します。

例

```
curl https://na1.salesforce.com/services/data/v26.0/ -H "Authorization: Bearer token"
```

リクエストボディの例

不要

JSON レスポンスボディの例

```

{
  "subjects" : "/services/data/v26.0/subjects",
  "licensing" : "/services/data/v26.0/licensing",
  "connect" : "/services/data/v26.0/connect",
  "search" : "/services/data/v26.0/search",
  "query" : "/services/data/v26.0/query",
  "tooling" : "/services/data/v26.0/tooling",
  "chatter" : "/services/data/v26.0/chatter",
  "recent" : "/services/data/v26.0/recent"
}

```

XML レスポンスボディの例

```

<?xml version="1.0" encoding="UTF-8"?>
<urls>
  <subjects>/services/data/v26.0/subjects</subjects>
  <licensing>/services/data/v26.0/licensing</licensing>
  <connect>/services/data/v26.0/connect</connect>
  <search>/services/data/v26.0/search</search>
  <query>/services/data/v26.0/query</query>
  <tooling>/services/data/v26.0/tooling</tooling>
  <chatter>/services/data/v26.0/chatter</chatter>
  <recent>/services/data/v26.0/recent</recent>
</urls>

```

オブジェクトのリストを取得する

組織で利用できるオブジェクトおよびログインユーザが利用できるオブジェクトをリストするには、[Describe Global](#) リソースを使用します。このリソースは、組織の文字コードとクエリで許可される最大バッチサイズも返します。

使用例

```
curl https://na1.salesforce.com/services/data/v26.0/subjects/ -H "Authorization: Bearer token"
```

リクエストボディの例

不要

レスポンスボディの例

```
{
  "encoding" : "UTF-8",
  "maxBatchSize" : 200,
  "subjects" : [ {
    "name" : "Account",
    "label" : "Account",
    "keyPrefix" : "001",
    "labelPlural" : "Accounts",
    "custom" : false,
    "layoutable" : true,
    "activateable" : false,
    "urls" : {
      "subject" : "/services/data/v26.0/subjects/Account",
      "describe" : "/services/data/v26.0/subjects/Account/describe",
      "rowTemplate" : "/services/data/v26.0/subjects/Account/{ID}"
    },
    "searchable" : true,
    "updateable" : true,
    "createable" : true,
    "deprecatedAndHidden" : false,
    "customSetting" : false,
    "deletable" : true,
    "feedEnabled" : true,
    "mergeable" : true,
    "queryable" : true,
    "replicateable" : true,
    "retrieveable" : true,
    "undeletable" : true,
    "triggerable" : true
  },
  ...
]
```

オブジェクトメタデータの使用

REST API を使用して、オブジェクトメタデータの基本情報を取得できます。

このセクションの例では、REST API リソースを使用して、オブジェクトメタデータ情報を取得します。オブジェクトメタデータ情報の変更または作成についての詳細は、『[メタデータAPI開発者ガイド](#)』を参照してください。

オブジェクトのメタデータを取得する

オブジェクトのメタデータを取得するには、[sObject Basic Information](#) リソースを使用します。

Account メタデータを取得する場合の使用例

```
curl https://na1.salesforce.com/services/data/v20.0/subjects/Account/ -H "Authorization: Bearer token"
```

Account メタデータを取得する場合のリクエストボディの例

不要

Account メタデータを取得する場合のレスポンスボディの例

```
{
  "objectDescribe" :
  {
    "name" : "Account",
    "updateable" : true,
    "label" : "Account",
    "keyPrefix" : "001",
    ...
    "replicateable" : true,
    "retrieveable" : true,
    "undeletable" : true,
    "triggerable" : true
  },
  "recentItems" :
  [
    {
      "attributes" :
      {
        "type" : "Account",
        "url" : "/services/data/v20.0/subjects/Account/001D000000INjVeIAL"
      },
      "Id" : "001D000000INjVeIAL",
      "Name" : "asdasdasd"
    },
    ...
  ]
}
```

項目名やメタデータを含む、オブジェクトの完全な説明を取得するには、「[オブジェクトのリストを取得する](#)」を参照してください。

オブジェクトの項目と他のメタデータを取得する

各項目、URL、および子リレーションに関する情報を含む、オブジェクトのすべてのメタデータを取得するには、[sObject Describe](#) リソースを使用します。

例

```
https://na1.salesforce.com/services/data/v20.0/Account/describe/ -H "Authorization: Bearer token"
```


リクエストボディの例

不要

レスポンスボディの例

```
{
  "name" : "Account",
  "fields" :
  [
    {
      "length" : 18,
      "name" : "Id",
      "type" : "id",
      "defaultValue" : { "value" : null },
      "updateable" : false,
      "label" : "Account ID",
      ...
    },
    ...
  ],

  "updateable" : true,
  "label" : "Account",
  "keyPrefix" : "001",
  "custom" : false,
  ...

  "urls" :
  {
    "uiEditTemplate" : "https://na1.salesforce.com/{ID}/e",
    "sobject" : "/services/data/v20.0/sobjects/Account",
    "uiDetailTemplate" : "https://na1.salesforce.com/{ID}",
    ...
  },

  "childRelationships" :
  [
    {
      "field" : "ParentId",
      "deprecatedAndHidden" : false,
      ...
    },
    ....
  ],

  "createable" : true,
  "customSetting" : false,
  ...
}
```

オブジェクトのメタデータの変更の取得

sObjectDescribe リソースおよび If-Modified-Since HTTP ヘッダーを使用して、オブジェクトのメタデータが変更されたかどうかを確認できます。

[sObject Describe](#) 要求で、If-Modified-Since ヘッダーが `EEE, dd MMM yyyy HH:mm:ss z` 形式の日付とともに指定されている場合、応答のメタデータは、オブジェクトのメタデータが指定の日付以降に変更された場合に

のみ返されます。指定の日付以降にメタデータが変更されていないと、レスポンスボディなしで 304 Not Modified 状況コードが返されます。

次の例では、2013 年 7 月 3 日以降に新しいカスタム項目などの Merchandise__c オブジェクトに行われた変更がないことを前提としています。

sObject Describe 要求の例

```
/services/data/v29.0/sobjects/Merchandise__c/describe
```

要求で使用する If-Modified-Since ヘッダーの例

```
If-Modified-Since: Wed, 3 Jul 2013 19:43:31 GMT
```

レスポンスボディの例

レスポンスボディは返されない

応答状況コードの例

```
HTTP/1.1 304 Not Modified  
Date: Fri, 12 Jul 2013 05:03:24 GMT
```

2013 年 7 月 3 日以降に Merchandise__c に変更があった場合は、レスポンスボディには Merchandise__c のメタデータが含まれます。例については、「[オブジェクトの項目と他のメタデータを取得する](#)」を参照してください。

レコードの操作

REST API を使用して組織のレコードを操作できます。

このセクションの例では、REST API リソースを使用してレコードの作成、取得、更新、削除とその他のレコード関連操作を実行します。

レコードを作成する

新規レコードを作成するには、[sObject Basic Information](#) リソースを使用します。要求データの必須項目値を指定し、リソースの POST メソッドを使用します。コールが成功すると、レスポンスボディに、作成されたレコードの ID が含まれます。

次の例では、newaccount.json に指定された項目値で新規 Account レコードを作成します。

新規 Account を作成する例

```
curl https://na1.salesforce.com/services/data/v20.0/sobjects/Account/ -H "Authorization: Bearer token" -H "Content-Type: application/json" -d @newaccount.json
```

新規 Account を作成する場合のリクエストボディ newaccount.json ファイルの例

```
{  
  "Name" : "Express Logistics and Transport"  
}
```

新規 Account が正常に作成された場合のレスポンスボディの例

```
{
  "id" : "001D000000IqhSLIAZ",
  "errors" : [ ],
  "success" : true
}
```

レコードを更新する

レコードを更新するには、[sObject Rows](#) リソースを使用します。要求データに更新するレコード情報を指定し、リソースの PATCH メソッドを使用して、特定のレコードの ID を指定し、そのレコードを更新します。1つのファイルに含まれるレコードは、同じオブジェクト種別である必要があります。

次の例では、Account 内の請求先市区郡の情報 (BillingCity) が更新されます。更新するレコード情報は patchaccount.json に指定されています。

Account オブジェクトを更新する例

```
curl https://na1.salesforce.com/services/data/v20.0/sobjects/Account/001D000000INjVe
-H "Authorization: Bearer token" -H "Content-Type: application/json" -d @patchaccount.json
-X PATCH
```

Account オブジェクトの項目を更新する場合のリクエストボディ patchaccount.json ファイルの例

```
{
  "BillingCity" : "San Francisco"
}
```

Account オブジェクトの項目を更新する場合のレスポンスボディの例

戻り値なし

エラー応答

「[状況コードとエラー応答](#)」(ページ 88)を参照してください。

次の例では、Java と HttpClient を使用し、REST API を使用してレコードを更新します。HttpClient に PatchMethod がないため、PostMethod が上書きされてメソッド名として「PATCH」が返されます。この例では、オブジェクト名とレコード ID が含まれるリソース URL が渡されていると想定します。

```
public static void patch(String url, String sid) throws IOException {
    PostMethod m = new PostMethod(url) {
        @Override public String getName() { return "PATCH"; }
    };

    m.setRequestHeader("Authorization", "OAuth " + sid);

    Map<String, Object> accUpdate = new HashMap<String, Object>();
    accUpdate.put("Name", "Patch test");
    ObjectMapper mapper = new ObjectMapper();
    m.setRequestBody(new StringRequestEntity(mapper.writeValueAsString(accUpdate),
    "application/json", "UTF-8"));

    HttpClient c = new HttpClient();
    int sc = c.executeMethod(m);
}
```

```

    System.out.println("PATCH call returned a status code of " + sc);
    if (sc > 299) {
        // deserialize the returned error message
        List<ApiError> errors = mapper.readValue(m.getResponseBodyAsStream(), new
TypeReference<List<ApiError>>() {} );
        for (ApiError e : errors)
            System.out.println(e.errorCode + " " + e.message);
    }
}

private static class ApiError {
    public String errorCode;
    public String message;
    public String [] fields;
}

```

任意の HTTP メソッド名の上書きまたは設定を許可しない HTTP ライブラリを使用している場合、POST 要求を送信して、クエリ文字列パラメータ `_HttpMethod` を介して HTTP メソッドを上書きすることができます。この PATCH の例では、PostMethod の行を上書きを使用しない行に置き換えることができます。

```
PostMethod m = new PostMethod(url + "?_HttpMethod=PATCH");
```

レコードを削除する

レコードを削除するには、[sObject Rows](#) リソースを使用します。レコード ID を指定し、リソースの DELETE メソッドを使用してレコードを削除します。

Account レコードを削除する場合の例

```
curl https://na1.salesforce.com/services/data/v20.0/subjects/Account/001D000000INjVe
-H "Authorization: Bearer token" -X DELETE
```

リクエストボディの例

不要

レスポンスボディの例

戻り値なし

レコードから項目値を取得する

レコードから項目値を取得するには、[sObject Rows](#) リソースを使用します。fields パラメータに取得する項目を指定し、リソースの GET メソッドを使用します。次の例では、Account から取引先番号 (AccountNumber) と請求先の郵便番号 (BillingPostalCode) を取得します。

Account レコードの項目から値を取得する例

```
curl https://na1.salesforce.com/services/data/v20.0/subjects/Account/001D000000INjVe
?fields=AccountNumber,BillingPostalCode -H "Authorization: Bearer token"
```

リクエストボディの例

不要

レスポンスボディの例

```
{
  "AccountNumber" : "CD656092",
  "BillingPostalCode" : "27215",
}
```

外部 ID を使用してレコードを取得する

[sObject Rows by External ID](#) リソースの GET メソッドを使用して、特定の外部 ID でレコードを取得できます。

次の例では、MerchandiseExtID__c 外部 ID 項目を持つ Merchandise__c カスタムオブジェクトがあると想定します。

外部 ID を使用して Merchandise__c レコードを取得する場合の使用例

```
curl
https://na1.salesforce.com/services/data/v20.0/subjects/Merchandise__c/MerchandiseExtID__c/123
-H "Authorization: Bearer token"
```

リクエストボディの例

不要

レスポンスボディの例

```
{
  "attributes" : {
    "type" : "Merchandise__c",
    "url" : "/services/data/v20.0/subjects/Merchandise__c/a00D00000008oWP8IAM"
  },
  "Id" : "a00D00000008oWP8IAM",
  "OwnerId" : "005D00000001KyEIIA0",
  "IsDeleted" : false,
  "Name" : "Example Merchandise",
  "CreateDate" : "2012-07-12T17:49:01.000+0000",
  "CreatedById" : "005D00000001KyEIIA0",
  "LastModifiedDate" : "2012-07-12T17:49:01.000+0000",
  "LastModifiedById" : "005D00000001KyEIIA0",
  "SystemModstamp" : "2012-07-12T17:49:01.000+0000",
  "Description__c" : "Merch with external ID",
  "Price__c" : 10.0,
  "Total_Inventory__c" : 100.0,
  "Distributor__c" : null,
  "MerchandiseExtID__c" : 123.0
}
```

外部 ID を使用してレコードを挿入/更新 (Upsert) する

指定された外部 ID 項目の値に基づいて、新規レコードを作成するか、既存のレコードを挿入/更新 (Upsert) するには、[sObject Rows by External ID](#) を使用します。

- 指定された値が存在しない場合、新しいレコードが作成されます。
- 指定された値のレコードが存在する場合、リクエストボディに指定された項目値が更新されます。

- 値が一意でない場合、REST API によって、一致するレコードのリストと共に HTTP 状況コード 300 が返されます。

以降のセクションでは、外部 ID リソースを使用して外部 ID でレコードを取得する方法とレコードを Upsert する方法を説明します。

新規レコードの Upsert

この例では、PATCH メソッドを使用して新規レコードを挿入します。外部 ID 項目「customExtIdField__c」がすでに Account に追加されていると想定します。また、customExtIdField 値が 11999 の Account レコードがまだ存在していないと想定します。

まだ存在していないレコードを Upsert する例

```
curl
https://na1.salesforce.com/services/data/v20.0/subjects/Account/customExtIdField__c/11999
-H "Authorization: Bearer token" -H "Content-Type: application/json" -d @newrecord.json
-X PATCH
```

JSON リクエストボディ newrecord.json ファイルの例

```
{
  "Name" : "California Wheat Corporation",
  "Type" : "New Customer"
}
```

応答

成功時の応答は次のとおりです。

```
{
  "id" : "001900000001pPvHAAU",
  "errors" : [ ],
  "success" : true
}
```

レスポンスボディは空です。新しいレコードが作成されると、HTTP 状況コード 201 が返されます。

エラー応答

外部 ID 項目が不正な場合は、次のような応答が返されます。

```
{
  "message" : "The requested resource does not exist",
  "errorCode" : "NOT_FOUND"
}
```

詳細は、「[状況コードとエラー応答](#)」(ページ 88)を参照してください。

既存のレコードの Upsert

この例では、PATCH メソッドを使用して既存のレコードを更新します。外部 ID 項目「customExtIdField__c」がすでに Account に追加されていて、customExtIdField 値が 11999 の Account レコードがすでに存在すると想定します。要求では updates.json を使用して更新する項目値を指定します。

既存のレコードを Upsert する例

```
curl
https://na1.salesforce.com/services/data/v20.0/subjects/Account/customExtIdField__c/11999
-H "Authorization: Bearer token" -H "Content-Type: application/json" -d @updates.json
-X PATCH
```

JSON リクエストボディ updates.json ファイルの例

```
{
  "BillingCity" : "San Francisco"
}
```

応答の例 (JSON)

既存のレコードが更新される場合、HTTP 状況コード 204 が返されます。

エラー応答

外部 ID 値が一意でない場合、HTTP 状況コード 300 が返され、さらにクエリに一致したレコードのリストが返されます。エラーについての詳細は、「[状況コードとエラー応答](#)」(ページ 88)を参照してください。

外部 ID 項目が存在しない場合、エラーメッセージとコードが返されます。

```
{
  "message" : "The requested resource does not exist",
  "errorCode" : "NOT_FOUND"
}
```

レコードの Upsert と外部 ID との関連付け

オブジェクトでリレーションを使用して他のオブジェクトを参照する場合、REST API を使用し、新規レコードの挿入/更新の両方と、さらに外部 ID を使用した別のオブジェクトの参照も行うことができます。

次の例では、新規レコードを作成し、外部 ID を介して親レコードに関連付けます。次の条件を想定します。

- Merchandise__c カスタムオブジェクトには、外部 ID 項目 MerchandiseExtID__c がある。
- Line_Item__c カスタムオブジェクトには、外部 ID 項目 LineItemExtID__c と、Merchandise__c へのリレーションがある。
- MerchandiseExtID__c 値が 123 の Merchandise__c レコードが存在する。
- LineItemExtID__c 値が 456 の Line_Item__c レコードは存在しない。これは作成され、Merchandise__c レコードに関連付けられるレコードです。

新規レコードを Upsert して関連オブジェクトを参照する例

```
curl
https://na1.salesforce.com/services/data/v25.0/subjects/Line_Item__c/LineItemExtID__c/456
-H "Authorization: Bearer token" -H "Content-Type: application/json" -d @new.json -X
PATCH
```

JSON リクエストボディ `new.json` ファイルの例

関連する Merchandise__c レコードは、Merchandise__c の外部 ID 項目を使用して参照されます。

```
{
  "Name" : "LineItemCreatedViaExtID",
  "Merchandise__r" :
  {
    "MerchandiseExtID__c" : 123
  }
}
```

応答の例 (JSON)

作成が成功すると、HTTP 状況コード 201 が返されます。

```
{
  "id" : "a02D00000006YUhrIAO",
  "errors" : [ ],
  "success" : true
}
```

エラー応答

外部 ID 値が一意でない場合、HTTP 状況コード 300 が返され、さらにクエリに一致したレコードのリストが返されます。エラーについての詳細は、「[状況コードとエラー応答](#)」(ページ 88)を参照してください。

外部 ID 項目が存在しない場合、エラーメッセージとコードが返されます。

```
{
  "message" : "The requested resource does not exist",
  "errorCode" : "NOT_FOUND"
}
```

既存のレコードの更新には次の方法を使用することもできます。たとえば、上記の例の Line_Item__c を作成した場合、別の要求を使用して関連する Merchandise__c の更新を試みることができます。

レコードを更新する例

```
curl
https://na1.salesforce.com/services/data/v25.0/subjects/Line_Item__c/LineItemExtID__c/456
-H "Authorization: Bearer token" -H "Content-Type: application/json" -d @updates.json
-X PATCH
```

JSON リクエストボディ `updates.json` ファイルの例

ここでは、MerchandiseExtID__c 値が 333 の別の Merchandise__c レコードが存在すると想定します。

```
{
  "Merchandise__r" :
  {
    "MerchandiseExtID__c" : 333
  }
}
```

応答の例 (JSON)

既存のレコードが更新される場合、HTTP 状況コード 204 が返されます。

リレーション種別が主従関係で、リレーションが親の変更を許可しないように設定されている場合、親の外部 ID を更新しようとすると、HTTP 状況コード 400 エラーとエラーコード `INVALID_FIELD_FOR_INSERT_UPDATE` が返されます。

レコードから添付ファイルコンテンツを取得する

特定のレコードの blob データを取得するには、[sObject Blob Retrieve](#) リソースを使用します。

次の例では、Attachment レコードの blob データを取得します。Attachment は、Case、Campaign、または添付ファイルを許可するその他のオブジェクトに関連付けることができます。

Attachment レコードの blob 本文を取得する例

```
curl
https://na1.salesforce.com/services/data/v20.0/subjects/Attachment/001D000000INjVe/body
-H "Authorization: Bearer token"
```

リクエストボディの例

不要

レスポンスボディの例

Attachment の本文コンテンツがバイナリ形式で返されます。返されたデータがバイナリであるため、応答のコンテンツタイプは JSON でも XML でもありません。

次の例では、Document レコードの blob データを取得します。

Document レコードの blob 本文を取得する例

```
curl
https://na1.salesforce.com/services/data/v20.0/subjects/Document/015D0000000NdJOIA0/body
-H "Authorization: Bearer token"
```

リクエストボディの例

不要

レスポンスボディの例

Document の本文コンテンツがバイナリ形式で返されます。返されたデータがバイナリであるため、応答のコンテンツタイプは JSON でも XML でもありません。

Blob データを挿入または更新する



メモ: この REST 機能は現在、パイロットプログラムで使用可能です。組織でのこの機能の有効化については、salesforce.com にお問い合わせください。

[sObject Basic Information](#) および [sObject Rows](#) REST リソースを使用して、Salesforce 標準オブジェクトの blob データを挿入または更新できます。最大 500 MB の任意の種類のファイルをアップロードできますが、MIME マルチパートコンテンツタイプ標準に準拠するマルチパートメッセージを使用する必要があります。詳細については、[WC3 標準](#)を参照してください。blob 項目を含む標準オブジェクトのファイルを挿入/更新できます。



メモ: 非マルチパートメッセージを使用して blob データを挿入/更新できますが、これを行う場合、テキストデータは 50 MB まで、base64 で符号化されたデータは 37.5 MB までに制限されます。

リクエストメッセージボディの最初のパートには、Description または Name などの非バイナリ項目データが含まれます。メッセージの 2 つ目のパートには、アップロードするファイルのバイナリデータが含まれます。

この後のセクションでは、マルチパートコンテンツタイプを使用して blob データを挿入/更新する JSON の例を示します。

- [新規 Document の挿入](#)
- [Document の更新](#)
- [ContentVersion の挿入](#)
- [マルチパートメッセージの考慮事項](#)

新規 Document の挿入

このセクションには、新規 Document を作成するための構文とコードが含まれます。ファイル自体のバイナリデータの他に、Description、Keywords、Name などの他の項目データも指定されています。



ヒント: 新規 Document を追加すると、[ドキュメント] タブで変更の結果を確認できます。

新規 Document を作成する例

```
curl https://na1.salesforce.com/services/data/v23.0/subjects/Document/ -H "Authorization: Bearer token" -H "Content-Type: multipart/form-data; boundary=\"boundary_string\"" --data-binary @newdocument.json
```

新規 Document を作成する場合のリクエストボディの例

このコードは、newdocument.json のコンテンツです。ここでは、簡潔にするために PDF コンテンツのバイナリデータは省略され、「Binary data goes here.」に置き換えられています。実際の要求にはバイナリコンテンツ全体が含まれます。

```
--boundary_string
Content-Disposition: form-data; name="entity_document";
Content-Type: application/json

{
  "Description" : "Marketing brochure for Q1 2011",
  "Keywords" : "marketing,sales,update",
  "FolderId" : "005D0000001GiU7",
  "Name" : "Marketing Brochure Q1",
  "Type" : "pdf"
}

--boundary_string
Content-Type: application/pdf
Content-Disposition: form-data; name="Body"; filename="2011Q1MktgBrochure.pdf"

Binary data goes here.

--boundary_string--
```

新規 Document を作成する場合のレスポンスボディの例

成功すると、新規 Document の ID が返されます。

```
{
  "id" : "015D0000000N3ZZIA0",
  "errors" : [ ],
  "success" : true
}
```

エラー応答の例

```
{
  "fields" : [ "FolderId" ],
  "message" : "Folder ID: id value of incorrect type: 005D0000001GiU7",
  "errorCode" : "MALFORMED_ID"
}
```

Document の更新

このセクションには、既存の Document を更新するための構文とコードが含まれます。ファイル自体のバイナリデータの他に、Name や Keywords などの他の項目データも更新されます。

Document オブジェクトの項目を更新する場合の使用例

```
curl https://na1.salesforce.com/services/data/v23.0/Document/015D0000000N3ZZIA0 -H
"Authorization: Bearer token" -H "Content-Type: multipart/form-data;
boundary=\"boundary_string\"" --data-binary @UpdateDocument.json -X PATCH
```

Document オブジェクトの項目を更新する場合のリクエストボディの例

このコードは、UpdateDocument.json ファイルのコンテンツです。ここでは、簡潔にするために PDF コンテンツのバイナリデータは省略され、「Updated document binary goes here.」に置き換えられています。実際の要求にはバイナリコンテンツ全体が含まれます。

```
--boundary_string
Content-Disposition: form-data; name="entity_content";
Content-Type: application/json

{
  "Name" : "Marketing Brochure Q1 - Sales",
  "Keywords" : "sales, marketing, first quarter"
}

--boundary_string
Content-Type: application/pdf
Content-Disposition: form-data; name="Body"; filename="2011Q1MktgBrochure.pdf"

Updated document binary data goes here.

--boundary_string--
```

Document オブジェクトの項目を更新場合のレスポンスボディの例

戻り値なし

エラー応答

「[状況コードとエラー応答](#)」(ページ 88)を参照してください。

ContentVersion の挿入

このセクションには、新規 ContentVersion を挿入するための構文とコードが含まれます。ファイル自体のバイナリデータの他に、ReasonForChange や PathOnClient などの他の項目も更新されます。ContentVersion は、必ず ContentDocument に関連付けられているため、このメッセージには ContentDocumentId が含まれます。



ヒント: ContentVersion オブジェクトは update をサポートしていません。したがって ContentVersion を更新することはできませんが、新しい ContentVersion を挿入できます。変更の結果は、[コンテンツ] タブで確認できます。

ContentVersion を挿入する場合の使用例

```
curl https://na1.salesforce.com/services/data/v23.0/subjects/ContentVersion -H
"Authorization: Bearer token" -H "Content-Type: multipart/form-data;
boundary=\"boundary_string\"" --data-binary @NewContentVersion.json
```

ContentVersion を挿入する場合のリクエストボディの例

このコードは、NewContentVersion.json ファイルのコンテンツです。ここでは、簡潔にするために PDF コンテンツのバイナリデータは省略され、「Binary data goes here.」に置き換えられています。実際の要求にはバイナリコンテンツ全体が含まれます。

```
--boundary_string
Content-Disposition: form-data; name="entity_content";
Content-Type: application/json

{
  "ContentDocumentId" : "069D00000000so2",
  "ReasonForChange" : "Marketing materials updated",
  "PathOnClient" : "Q1 Sales Brochure.pdf"
}

--boundary_string
Content-Type: application/octet-stream
Content-Disposition: form-data; name="VersionData"; filename="Q1 Sales Brochure.pdf"

Binary data goes here.

--boundary_string--
```

ContentVersion を挿入場合のレスポンスボディの例

```
{
  "id" : "068D00000000pg0IAQ",
  "errors" : [ ],
  "success" : true
}
```

ContentVersion を挿入した場合のエラー応答

「[状況コードとエラー応答](#)」(ページ 88)を参照してください。

マルチパートメッセージの考慮事項

blob データを挿入/更新するときの、マルチパートメッセージの形式に関するいくつかの考慮事項を次に示します。

境界文字列

- マルチパートメッセージの各パートを区分します。
- マルチパートコンテンツタイプで必須です。
- 70 文字まで入力できます。
- メッセージパートのどの部分にも出現しない文字列値である必要があります。
- 最初の境界文字列には、2つのハイフン (--) をプレフィックスとして使用する必要があります。
- 最後の境界文字列には、2つのハイフン (--) をポストフィックスとして使用する必要があります。

Content-Disposition ヘッダー

- 各メッセージパートで必須です。
- 値は `form-data` であり、`name` 属性が必要です。
 - ◇ 非バイナリのメッセージパートでは、`name` 属性に任意の値を使用できます。
 - ◇ バイナリのメッセージパートでは、`name` 属性に、バイナリデータを含むオブジェクト項目の名前が含まれている必要があります。新規 Document を追加した前の例では、ファイルを含むバイナリ項目の名前は「Body」です。
- バイナリのメッセージパートには、ローカルファイルの名前を表す `filename` 属性が必要です。

Content-Type ヘッダー

- 各メッセージパートで必須です。
- 非バイナリのメッセージパートでサポートされているコンテンツタイプは、`application/json` と `application/xml` です。
- バイナリのメッセージパートの `Content-Type` ヘッダーには、任意の値を使用できます。

改行

メッセージパートのヘッダーとそのパートのデータの間に、改行が必要です。コード例で示されるとおり、`Content-Type` ヘッダーや `Content-Disposition` ヘッダーと、JSON または XML の間に改行が必要です。バイナリのパートでは、`Content-Type` ヘッダーや `Content-Disposition` ヘッダーとバイナリデータの間に改行が必要です。

特定の期間に削除されたレコードのリストの取得

指定されたオブジェクトの削除されたレコードのリストを取得するには、[sObject Get Deleted](#) リソースを使用します。特定のオブジェクトのレコードが削除された日時の範囲を指定します。削除されたレコードは削除ログ (定期的に消去される) に書き込まれ、sObject 行、クエリなどのほとんどの操作対象から除外されます (ただし、QueryAll では削除されたレコードが結果に含まれます)。

2013 年 5 月 5 日～2013 年 5 月 10 日に削除された Merchandise__c レコードのリストを取得する場合の使用例

```
/services/data/v29.0/objects/Merchandise__c/deleted/  
?start=2013-05-05T00:00:00+00:00&end=2013-05-10T00:00:00+00:00
```

リクエストボディの例

不要

JSON レスポンスボディの例

```
{
  "deletedRecords" :
  [
    {
      "id" : "a00D0000008pQRAIA2",
      "deletedDate" : "2013-05-07T22:07:19.000+0000"
    }
  ],
  "earliestDateAvailable" : "2013-05-03T15:57:00.000+0000",
  "latestDateCovered" : "2013-05-08T21:20:00.000+0000"
}
```

XML レスポンスボディの例

```
<?xml version="1.0" encoding="UTF-8"?>
<Merchandise__c>
  <deletedRecords>
    <deletedDate>2013-05-07T22:07:19.000Z</deletedDate>
    <id>a00D0000008pQRAIA2</id>
  </deletedRecords>
  <earliestDateAvailable>2013-05-03T15:57:00.000Z</earliestDateAvailable>
  <latestDateCovered>2013-05-08T21:20:00.000Z</latestDateCovered>
</Merchandise__c>
```

特定の期間に更新されたレコードのリストの取得

指定されたオブジェクトの更新 (変更または追加) されたレコードのリストを取得するには、[sObject Get Updated](#) リソースを使用します。特定のオブジェクトのレコードが更新された日時の範囲を指定します。

2013 年 5 月 6 日～2013 年 5 月 10 日に更新された Merchandise__c レコードのリストを取得する場合の使用例

```
/services/data/v29.0/subjects/Merchandise__c/updated/
?start=2013-05-06T00:00:00+00:00&end=2013-05-10T00:00:00+00:00
```

リクエストボディの例

不要

JSON レスポンスボディの例

```
{
  "ids" :
  [
    "a00D0000008pQR5IAM",
    "a00D0000008pQRGIA2",
    "a00D0000008pQRFIA2"
  ],
  "latestDateCovered" : "2013-05-08T21:20:00.000+0000"
}
```

XML レスポンスボディの例

```
<?xml version="1.0" encoding="UTF-8"?>
<Merchandise__c>
  <ids>a00D00000008pQR5IAM</ids>
  <ids>a00D00000008pQRGIA2</ids>
  <ids>a00D00000008pQRFIA2</ids>
  <latestDateCovered>2013-05-08T21:20:00.000Z</latestDateCovered>
</Merchandise__c>
```

検索とクエリの使用

REST API を使用して、データに対して複雑な検索やクエリを実行できます。

このセクションの例では、REST API リソースを使用して、Salesforce Object Search Language (SOSL) および Salesforce Object Query Language (SOQL) を使用したレコードの検索やクエリを実行します。SOSL および SOQL についての詳細は、[『Force.com SOQL and SOSL Reference』](#)を参照してください。

SOQL クエリを実行する

すべての結果を1つの応答で返すか、または必要に応じて、結果の一部と、残りの結果を取得するために使用する識別子を返す SOQL クエリを実行するには、[Query](#) リソースを使用します。

次のクエリは、すべての Account レコードを対象に name 項目の値を要求します。

クエリを実行する場合の使用例

```
curl https://na1.salesforce.com/services/data/v20.0/query/?q=SELECT+name+from+Account
-H "Authorization: Bearer token"
```

クエリを実行する場合のリクエストボディの例

不要

クエリを実行場合のレスポンスボディの例

```
{
  "done" : true,
  "totalSize" : 14,
  "records" :
  [
    {
      "attributes" :
      {
        "type" : "Account",
        "url" : "/services/data/v20.0/subjects/Account/001D000000IRFmaIAH"
      },
      "Name" : "Test 1"
    },
    {
      "attributes" :
      {
        "type" : "Account",
        "url" : "/services/data/v20.0/subjects/Account/001D000000IomazIAB"
      },
      "Name" : "Test 2"
    }
  ],
}
```

```

    ...
  ]
}

```

SOQL クエリの残りの結果の取得

最初のクエリで結果の一部のみを返す場合、応答の最後に `nextRecordsUrl` という項目が含まれます。たとえば、クエリの最後に次の属性があるとします。

```
"nextRecordsUrl" : "/services/data/v20.0/query/01gD0000002HU6KIAW-2000"
```

この場合、レコードの次のバッチを要求し、すべてのレコードが取得されるまでこの操作を繰り返します。これらの要求は `nextRecordsUrl` を使用し、パラメータを含みません。

クエリの残りの結果を取得する場合の使用例

```
curl https://na1.salesforce.com/services/data/v20.0/query/01gD0000002HU6KIAW-2000 -H
"Authorization: Bearer token"
```

クエリの残りの結果を取得する場合のリクエストボディの例

不要

クエリの残りの結果を取得する場合のレスポンスボディの例

```

{
  "done" : true,
  "totalSize" : 3214,
  "records" : [...]
}

```

削除された項目を含む SOQL クエリを実行する

`merge` または `delete` で削除されたレコードの情報を含む SOQL クエリを実行するには、[QueryAll](#) リソースを使用します。Query リソースでは削除された項目が自動的に除外されるため、Query ではなく QueryAll を使用します。

次のクエリは、削除された `Merchandise__c` レコードが 1 つある組織で、削除されたすべての `Merchandise__c` レコードを対象に `Name` 項目の値を要求します。同じクエリに QueryAll ではなく Query を使用した場合、レコードは返されません。これは、Query では削除されたレコードがすべて結果セットから自動的に除外されるためです。

削除された Merchandise__c レコードのクエリを実行する場合の使用例

```
/services/data/v29.0/queryAll/?q=SELECT+Name+from+Merchandise__c+WHERE+isDeleted+=+TRUE
```

クエリを実行する場合のリクエストボディの例

不要

クエリを実行する場合のレスポンスボディの例

```
{
  "done" : true,
  "totalSize" : 1,
  "records" :
  [
    {
      "attributes" :
      {
        "type" : "Merchandise__c",
        "url" : "/services/data/v29.0/objects/Merchandise__c/a00D00000008pQRAIX2"
      },
      "Name" : "Merchandise 1"
    },
  ]
}
```

SOQL クエリの残りの結果の取得

最初のクエリで結果の一部のみを返す場合、応答の最後に `nextRecordsUrl` という項目が含まれます。たとえば、クエリの最後に次の属性があるとして。

```
"nextRecordsUrl" : "/services/data/v29.0/query/01gD00000002HU6KIAW-2000"
```

この場合、レコードの次のバッチを要求し、すべてのレコードが取得されるまでこの操作を繰り返します。これらの要求は `nextRecordsUrl` を使用し、パラメータを含みません。

`nextRecordsUrl` の URL に `query` が指定されている場合でも、最初の QueryAll 要求の残りの結果が提供されます。残りの結果には、最初のクエリに一致した削除されたレコードが含まれます。

残りの結果を取得する場合の使用例

```
/services/data/v29.0/query/01gD00000002HU6KIAW-2000
```

残りの結果を取得する場合のリクエストボディの例

不要

残りの結果を取得する場合のレスポンスボディの例

```
{
  "done" : true,
  "totalSize" : 3214,
  "records" : [...]
}
```

文字列を検索する

SOSL 検索を実行するには、[Search](#) リソースを使用します。

次の例では、`{test}` の SOSL 検索を実行します。この例の検索文字列は URL 符号化されている必要があります。

使用例

```
curl https://na1.salesforce.com/services/data/v20.0/search/?q=FIND+%7Btest%7D -H
"Authorization: Bearer token"
```

リクエストボディの例

不要

レスポンスボディの例

```
[
  {
    "attributes" :
    {
      "type" : "Account",
      "url" : "/services/data/v20.0/subjects/Account/001D000000IqhSLIAZ"
    },
    "Id" : "001D000000IqhSLIAZ"
  },
  {
    "attributes" :
    {
      "type" : "Account",
      "url" : "/services/data/v20.0/subjects/Account/001D000000IomazIAB"
    },
    "Id" : "001D000000IomazIAB"
  }
]
```

デフォルトの検索範囲と検索順序の取得

[Search Scope and Order](#) リソースを使用して、ユーザの検索結果ページの固定表示オブジェクトを含め、ログインユーザのデフォルトの検索範囲と検索順序を取得します。

次の例では、ログインユーザのデフォルトのグローバル検索範囲は、サイト、アイデア、ケース、商談、取引先、およびユーザオブジェクトがレスポンスボディに返される順序で構成されます。

使用例

```
curl https://na1.salesforce.com/services/data/v26.0/search/scopeOrder -H "Authorization:
Bearer token"
```

リクエストボディの例

不要

レスポンスボディの例

```
[
  {
    "type": "Site",
    "url": "/services/data/v26.0/subjects/Site/describe"
  },
  {
    "type": "Idea",
    "url": "/services/data/v26.0/subjects/Idea/describe"
  }
]
```

```

    },
    {
      "type": "Case",
      "url": "/services/data/v26.0/subjects/Case/describe"
    },
    {
      "type": "Opportunity",
      "url": "/services/data/v26.0/subjects/Opportunity/describe"
    },
    {
      "type": "Account",
      "url": "/services/data/v26.0/subjects/Account/describe"
    },
    {
      "type": "User",
      "url": "/services/data/v26.0/subjects/User/describe"
    }
  ]
}

```

オブジェクトの検索結果レイアウトの取得

[Search Result Layouts](#) リソースを使用して、クエリ文字列で指定された各オブジェクトの検索結果レイアウトの設定を取得します。

使用例

```

curl
https://na1.salesforce.com/services/data/v28.0/searchlayout/?q=Account,Contact,Lead,Asset
"Authorization: Bearer token"

```

リクエストボディの例

不要

レスポンスボディの例

```

[ { "label" : "Search Results",
  "limitRows" : 25,
  "searchColumns" : [ { "field" : "Account.Name",
    "format" : null,
    "label" : "Account Name",
    "name" : "Name"
  },
  { "field" : "Account.Site",
    "format" : null,
    "label" : "Account Site",
    "name" : "Site"
  },
  { "field" : "Account.Phone",
    "format" : null,
    "label" : "Phone",
    "name" : "Phone"
  },
  { "field" : "User.Alias",
    "format" : null,
    "label" : "Account Owner Alias",
    "name" : "Owner.Alias"
  }
  ]
},
{ "label" : "Search Results",

```

```

"limitRows" : 25,
"searchColumns" : [ { "field" : "Contact.Name",
  "format" : null,
  "label" : "Name",
  "name" : "Name"
},
{ "field" : "Account.Name",
  "format" : null,
  "label" : "Account Name",
  "name" : "Account.Name"
},
{ "field" : "Account.Site",
  "format" : null,
  "label" : "Account Site",
  "name" : "Account.Site"
},
{ "field" : "Contact.Phone",
  "format" : null,
  "label" : "Phone",
  "name" : "Phone"
},
{ "field" : "Contact.Email",
  "format" : null,
  "label" : "Email",
  "name" : "Email"
},
{ "field" : "User.Alias",
  "format" : null,
  "label" : "Contact Owner Alias",
  "name" : "Owner.Alias"
}
]
},
{ "label" : "Search Results",
  "limitRows" : 25,
  "searchColumns" : [ { "field" : "Lead.Name",
    "format" : null,
    "label" : "Name",
    "name" : "Name"
  },
  { "field" : "Lead.Title",
    "format" : null,
    "label" : "Title",
    "name" : "Title"
  },
  { "field" : "Lead.Phone",
    "format" : null,
    "label" : "Phone",
    "name" : "Phone"
  },
  { "field" : "Lead.Company",
    "format" : null,
    "label" : "Company",
    "name" : "Company"
  },
  { "field" : "Lead.Email",
    "format" : null,
    "label" : "Email",
    "name" : "Email"
  },
  { "field" : "Lead.Status",
    "format" : null,
    "label" : "Lead Status",
    "name" : "toLabel(Status)"
  },
  { "field" : "Name.Alias",
    "format" : null,
    "label" : "Owner Alias",
    "name" : "Owner.Alias"
  }
]
}

```

```

    }
  ],
}
]

```

最近参照した情報の操作

REST API を使用して、最近参照したレコードに関する情報を取得および設定できます。

このセクションの例では、REST API の [Query](#) リソースおよび [Recently Viewed](#) リソースを使用して最近参照したレコード情報をプログラムで取得および更新します。

最近参照したレコードの表示

最近参照したデータとしてレコードをマーク

最近参照したレコードの表示

最近参照したレコードのリストを取得するには、[Recently Viewed Items](#) リソースを使用します。

最近参照したレコードのうち最近の 2 件を取得する場合の使用例

```
/services/data/v28.0/recent/?limit=2
```

リクエストボディの例

不要

レスポンスボディの例

```

{
  "attributes" :
  {
    "type" : "Account",
    "url" : "/services/data/v28.0/subjects/Account/a06U000000CelH0IAJ"
  },
  "Id" : "a06U000000CelH0IAJ",
  "Name" : "Acme"
},
{
  "attributes" :
  {
    "type" : "Opportunity",
    "url" : "/services/data/v28.0/subjects/Opportunity/a06U000000CelGvIAJ"
  },
  "Id" : "a06U000000CelGvIAJ",
  "Name" : "Acme - 600 Widgets"
}

```

最近参照したデータとしてレコードをマーク

REST API を使用して、最近参照したデータとしてレコードをマークするには、FOR VIEW 句または FOR REFERENCE 句を指定して [Query](#) リソースを使用します。レコードを最近参照したデータとしてマークし、レコードが参照された日時などの情報が正しく設定されていることを確認するには、SOQL を使用します。

FOR VIEW は、モバイルアプリケーションなどのカスタムインターフェースまたはカスタムページからレコードが参照された場合に、Salesforce に通知するために使用します。レコードがカスタムインターフェースから参照されている場合は、FOR REFERENCE を使用します。レコードは、関連レコードが表示されるたびに参照されます。詳細については、『*Force.com SOQL および SOSL リファレンス*』の「FOR VIEW」および「FOR REFERENCE」を参照してください。

1つの取引先レコードを最近参照したデータとしてマークするクエリを実行する場合の使用例

```
/services/data/v28.0/query/?q=SELECT+Name+FROM+Account+LIMIT+1+FOR+VIEW
```

クエリを実行する場合のリクエストボディの例

不要

クエリを実行する場合のレスポンスボディの例

```
{
  "done" : true,
  "totalSize" : 1,
  "records" :
  [
    {
      "attributes" :
      {
        "type" : "Account",
        "url" : "/services/data/v28.0/objects/Account/001D000000IRFmaIAH"
      },
      "Name" : "Acme"
    },
  ],
}
```

ユーザパスワードの管理

REST API を使用して、組織のユーザのログイン情報を管理できます。

このセクションの例では、REST API リソースを使用して、パスワードの設定やリセットなど、ユーザパスワードを管理します。

ユーザパスワードを管理する

ユーザパスワードの設定やリセット、パスワードに関する情報の取得を行うには、[sObject User Password](#) リソースを使用します。パスワードの有効期限の状況を取得するには HTTP GET メソッド、パスワードを設定するには HTTP POST メソッド、パスワードをリセットするには HTTP DELETE メソッドを使用します。

関連付けられたセッションには、特定のユーザパスワード情報へのアクセス権が必要です。セッションに適切な権限がない場合、これらのメソッドから HTTP エラー応答 403 が返されます。

これらのメソッドは、ユーザとセルフサービスユーザの両方に提供されています。セルフサービスユーザのパスワードの管理では、REST API URL に、User の代わりに SelfServiceUser を使用します。

次に、ユーザの現在のパスワード有効期限の状況を取得する例を示します。

現在のパスワード有効期限の状況を取得する場合の使用例

```
curl
https://na1.salesforce.com/services/data/v25.0/subjects/User/005D00000001KyEIIA0/password
-H "Authorization: Bearer token"
```

現在のパスワード有効期限の状況を取得する場合のリクエストボディの例

不要

現在のパスワード有効期限の状況を取得する場合のレスポンスボディの例 (JSON)

```
{
  "isExpired" : false
}
```

現在のパスワード有効期限の状況を取得する場合のレスポンスボディの例 (XML)

```
<Password>
  <isExpired>false</isExpired>
</Password>
```

セッションの権限が不十分な場合のエラー応答の例

```
{
  "message" : "You do not have permission to view this record.",
  "errorCode" : "INSUFFICIENT_ACCESS"
}
```

次に、特定のユーザのパスワードを変更する例を示します。

ユーザパスワードを変更する場合の使用例

```
curl
https://na1.salesforce.com/services/data/v25.0/subjects/User/005D00000001KyEIIA0/password
-H "Authorization: Bearer token" -H "Content-Type: application/json" -d @newpwd.json
-X POST
```

ファイル newpwd.json のコンテンツ

```
{
  "NewPassword" : "myNewPassword1234"
}
```

ユーザパスワードを変更する場合のレスポンスボディの例

パスワードが正しく変更された場合のレスポンスボディはありません。HTTP 状況コード 204 が返されます。

新規パスワードが組織のパスワード要件を満たしていない場合のエラー応答の例

```
{
  "message" : "Your password must have a mix of letters and numbers.",
  "errorCode" : "INVALID_NEW_PASSWORD"
}
```

最後に、ユーザパスワードのリセットの例を示します。

ユーザパスワードをリセットする場合の使用例

```
curl
https://na1.salesforce.com/services/data/v25.0/subjects/User/005D00000001KyEIIA0/password
-H "Authorization: Bearer token" -X DELETE
```

ユーザパスワードをリセットする場合のリクエストボディの例

不要

ユーザパスワードをリセットする場合のレスポンスボディの例 (JSON)

```
{
  "NewPassword" : "2sv0xHAuM"
}
```

ユーザパスワードをリセットする場合のレスポンスボディの例 (XML)

```
<Result>
  <NewPassword>2sv0xHAuM</NewPassword>
</Result>
```


REST API リファレンス

第4章

リファレンス

次の表に、APIでサポートされている REST リソースをリストし、それぞれのリソースについて簡単に説明します。それぞれの場合で、リソースの URI は、認証サービスから取得するベース URI (<http://domain/services/data>) に続きます。**domain** は、使用している Salesforce インスタンス、または [カスタムドメイン](#) です。たとえば、バージョン 20.0 の Account オブジェクトに関する基本情報を取得する場合、<http://na1.salesforce.com/services/data/v20.0/subjects/Account/> となります。

コール名をクリックすると、構文、使用方法、各コールの詳細情報を確認できます。

リソース名	URI	説明
Versions	/	バージョン、表示ラベル、および各バージョンのルートへのリンクなど、現在使用可能な各 Salesforce バージョンの概要情報をリストします。
Resources by Version	/vXX.X/	リソース名およびURIを含む、指定された API バージョンで使用可能なリソースをリストします。
Describe Global	/vXX.X/subjects/	組織のデータで使用可能なオブジェクトとそのメタデータをリストします。
sObject Basic Information	/vXX.X/subjects/ sObject /	指定されたオブジェクトの個別のメタデータを説明します。特定のオブジェクトの新規レコードの作成にも使用できます。
sObject Describe	/vXX.X/subjects/ sObject /describe/	指定されたオブジェクトのすべてのレベルで、個別のメタデータを完全に説明します。
sObject Get Deleted	/vXX.X/subjects/ sObject /deleted/ ?start= startDateAndTime &end= endDateAndTime	指定されたオブジェクトについて、特定の期間内に削除された個々のレコードのリストを取得します。
sObject Get Updated	/vXX.X/subjects/ sObject /updated/ ?start= startDateAndTime &end= endDateAndTime	指定されたオブジェクトに対して指定された期間内に更新された (追加または変更さ

リソース名	URI	説明
		れた) 個別のレコードのリストを取得します。
sObject Rows	/vXX.X/subjects/ <i>sObject</i> / <i>id</i> /	指定されたオブジェクト ID に基づいてレコードにアクセスします。レコードを取得、更新、または削除します。このリソースは、項目値の取得にも使用できます。
sObject Rows by External ID	/vXX.X/subjects/ <i>sObject</i> / <i>fieldName</i> / <i>fieldValue</i>	指定された外部 ID 項目の値に基づいて、新しいレコードを作成するか、既存のレコードを更新(レコードを Upsert) します。
sObject CompactLayouts	/vXX.X/subjects/ <i>Object</i> /describe/compactLayouts/	特定のオブジェクトのコンパクトレイアウトのリストを返します。
sObject Layouts	/vXX.X/subjects/global/describe/layouts/ /vXX.X/subjects/ <i>object</i> /describe/layouts/	パブリッシャーアクションを含む、レイアウトおよび説明のリストを返します。
sObject Blob Retrieve	/vXX.X/subjects/ <i>sObject</i> / <i>id</i> / <i>blobField</i>	個別のレコードから指定された blob 項目を取得します。
sObject Quick Actions	/vXX.X/subjects/ <i>object</i> /quickActions/ /vXX.X/subjects/ <i>object</i> /quickActions/{アクション名} /vXX.X/subjects/ <i>object</i> /quickActions/{アクション名}/describe/ services/data/vXX.X/subjects/ <i>object</i> /quickActions/{アクション名}/defaultValues/ vXX.X/subjects/ <i>object</i> /quickActions/{アクション名}/defaultValues/{親 ID}	パブリッシャーアクションおよび詳細のリストを返します。
sObject User Password	/vXX.X/subjects/User/ <i>ユーザ ID</i> /password /vXX.X/subjects/SelfServiceUser/ <i>セルフサービスユーザ ID</i> /password	ユーザパスワードを設定またはリセットしたり、ユーザパスワードに関する情報を取得したりします。
AppMenu	/vXX.X/appMenu/AppSwitcher/ /vXX.X/appMenu/Salesforce1/	Salesforce アプリケーションドロップダウンメニューまたは Salesforce1 ナビゲーションメニューの項目のリストを返します。
FlexiPage	/vXX.X/flexiPage/ <i>フレキシブルページの ID</i>	フレキシブルページとその詳細のリストを返します。返される情報には、フレキシブルページの範囲、各範囲内のコンポーネント、各コンポーネントのプロパティ、および関連付けられた QuickActions が含まれます。
Query	/vXX.X/query/?q= <i>soql</i>	指定された SOQL クエリを実行します。

リソース名	URI	説明
QueryAll	/vXX.X/queryAll/?q= soql	指定された SOQL クエリを実行します。結果には削除されたレコード、マージされたレコード、およびアーカイブ済みレコードが含まれる場合があります。
Quick Actions	/vXX.X/quickActions/	グローバルパブリッシャーアクションとその種別のリスト、および Chatter フィードに表示されるカスタム項目とオブジェクトを返します。
Search	/vXX.X/search/?s= sosl	指定された SOSL 検索を実行します。検索文字列は URL 符号化されている必要があります。
Search Scope and Order	/vXX.X/search/scopeOrder	ログインユーザのデフォルトのグローバル検索範囲内にあるオブジェクトの順序付きリストを返します。グローバル検索は、操作するオブジェクトとそれら进行操作する頻度を追跡し、それに基づいて検索結果を編成します。最もよく使用されるオブジェクトは、リストの最上部に表示されます。
Search Result Layouts	/vXX.X/searchlayout/?q= カンマで区切られたオブジェクトのリスト	クエリ文字列に含まれるオブジェクトの検索結果レイアウトに関する情報を返します。このコールでは、検索結果ページに列として表示される項目のリスト、最初のページに表示される行数、および検索結果ページで使用されるラベルがオブジェクトごとに返されます。
Recently Viewed Items	/vXX.X/recent	現在のユーザが表示または参照した、最近参照された項目を取得します。
Themes	/vXX.X/theme	Salesforce アプリケーションのテーマで使用するアイコンと色のリストを取得します。

Versions

バージョン、表示ラベル、および各バージョンのルートへのリンクなど、現在使用可能な各 Salesforce バージョンの概要情報をリストします。

URI

/

形式

JSON、XML

HTTP メソッド

GET

認証

なし

パラメータ

なし

例

「[使用可能な REST API バージョンをリストする](#)」(ページ 31)を参照してください。

Resources by Version

リソース名および URI を含む、指定された API バージョンで使用可能なリソースをリストします。

URI`/vXX.X/`**形式**

JSON、XML

HTTP メソッド

GET

認証Authorization: Bearer ***token*****パラメータ**

なし

例

「[使用可能な REST リソースをリストする](#)」(ページ 32)を参照してください。

Describe Global

組織のデータで使用可能なオブジェクトとそのメタデータをリストします。さらに、組織の文字コードとクエリで許可される最大バッチサイズを返します。文字コードについての詳細は、「[国際化と文字コード](#)」を参照してください。

URI`/vXX.X/subjects/`**形式**

JSON、XML

HTTP メソッド

GET

認証

Authorization: Bearer **token**

パラメータ

不要

例

「[オブジェクトのリストを取得する](#)」(ページ 32)を参照してください。

エラー応答

「[状況コードとエラー応答](#)」(ページ 88)を参照してください。

sObject Basic Information

指定されたオブジェクトの個別のメタデータを説明します。特定のオブジェクトの新規レコードの作成にも使用できます。たとえば、これは、GET メソッドを使用した Account オブジェクトのメタデータの取得や、POST メソッドを使用した新規 Account オブジェクトの作成に使用できます。

URI

/vXX.X/subjects/**sObjectName**/

形式

JSON、XML

HTTP メソッド

GET、POST

認証

Authorization: Bearer **token**

パラメータ

不要

例

- オブジェクトのメタデータを取得する例は、「[オブジェクトのメタデータを取得する](#)」(ページ 33)を参照してください。
- POST を使用した新規レコードを作成する例は、「[レコードを作成する](#)」(ページ 36)を参照してください。
- レコードの blob データを指定して新規レコードを作成する例は、「[Blob データを挿入または更新する](#)」(ページ 43)を参照してください。

sObject Describe

指定されたオブジェクトのすべてのレベルで、個別のメタデータを完全に説明します。たとえば、これは、Account オブジェクトの項目、URL、および子リレーションを取得するために使用できます。

If-Modified-Since ヘッダーは、このリソースでは `EEE, dd MMM yyyy HH:mm:ss z` という日付形式で使用できます。このヘッダーが使用される場合、指定の日付以降にオブジェクトメタデータが変更されていないと、レスポンスボディなしで `304 Not Modified` 状況コードが返されます。

URI

`/vXX.X/subjects/sObjectName/describe/`

形式

JSON、XML

HTTP メソッド

GET

認証

Authorization: Bearer **token**

パラメータ

不要

例

「[オブジェクトの項目と他のメタデータを取得する](#)」(ページ34)を参照してください。If-Modified-Since HTTPヘッダーの使用例は、「[オブジェクトのメタデータの変更の取得](#)」(ページ35)を参照してください。

sObject Get Deleted

指定されたオブジェクトについて、特定の期間内に削除された個々のレコードのリストを取得します。sObject Get Deleted は、API バージョン 29.0 以降で使用できます。

このリソースは、データ複製アプリケーションで一般的に使用されます。次の考慮事項に注意してください。

- 削除されたレコードは、このリソースからアクセス可能な削除ログに出力されます。2時間ごとに実行されるバックグラウンドプロセスは、削除ログのレコード数が制限を超えた場合、削除ログに書き込まれてから2時間以上経過したレコードを消去します。最も古いレコードから順に、削除ログが制限を下回るまで消去を行います。大量の削除ログによる Salesforce のパフォーマンス上の問題を防ぐためにこの処理を行います。
- 削除されたレコードに関する情報は、現在のセッションのユーザにそれらのレコードへのアクセス権がある場合にのみ返されます。
- コールが実行された日から 15 日以内の結果が返されます (管理者がごみ箱の中身を消去した場合、期間が短くなる場合があります)。

データ複製およびデータ複製の制限についての詳細は、『[SOAP API 開発者ガイド](#)』の「データ複製」を参照してください。

URI

`/vXX.X/subjects/sObjectName/deleted/?start=startDateAndTime&end=endDateAndTime`

形式

JSON、XML

HTTP メソッド

GET

認証

Authorization: Bearer **token**

パラメータ

パラメータ	説明
start	データを取得する期間の開始日時 (ローカル時間ではなく協定世界時 (UTC))。API は、指定された dateTime 値の秒の値を切り捨てます (たとえば、12:30:15 は 12:30:00 UTC となります)。日時は、ISO 8601 形式 (YYYY-MM-DDThh:mm:ss+hh:mm) で指定する必要があります。start の日付/時間値は、end の値より過去の日時でなければなりません。
end	データを取得する期間の終了日時 (ローカル時間ではなく協定世界時 (UTC))。API は、指定された dateTime 値の秒の値を切り捨てます (たとえば、12:35:15 は 12:35:00 UTC となります)。日時は、ISO 8601 形式 (YYYY-MM-DDThh:mm:ss+hh:mm) で指定する必要があります。

応答形式

プロパティ	型	説明
deletedRecords	array	要求で指定された開始日と終了日を満たす削除されたレコードの配列。各エントリには、レコード ID と協定世界時 (UTC) タイムゾーンを使用した ISO 8601 形式でそのレコードが削除された日時が含まれています。
earliestDateAvailable	String	最後に物理的に削除されたオブジェクトの ISO 8601 形式のタイムスタンプ (ローカル時間ではなく協定世界時 (UTC) タイムゾーン)。
latestDateCovered	String	要求の対象となる最終日の ISO 8601 形式のタイムスタンプ (ローカル時間ではなく協定世界時 (UTC) タイムゾーン)。

例

削除された項目のリストを取得する例は、[「特定の期間に削除されたレコードのリストの取得」](#) (ページ 47)を参照してください。

sObject Get Updated

指定されたオブジェクトに対して指定された期間内に更新された (追加または変更された) 個別のレコードのリストを取得します。sObject Get Updated は、API バージョン 29.0 以降で使用できます。

このリソースは、データ複製アプリケーションで一般的に使用されます。次の考慮事項に注意してください。

- ・ コールが実行された日から 30 日以内の結果が返されます。
- ・ クライアントアプリケーションは、適切な権限が付与されている場合、任意のオブジェクトを複製できます。たとえば、組織のすべてのデータを複製するには、クライアントアプリケーションは指定されたオブジェク

トの「すべてのデータの参照」アクセス権限を持ってログインしなければなりません。同様に、オブジェクトはそのユーザの共有ルールに含まれていなければなりません。

- このリソースから返される ID は、200,000 件までに制限されています。200,000 件以上の ID が返された場合、EXCEEDED_ID_LIMIT が返されます。開始日と終了日の期間を短くすることでこのエラーを回避できます。

データ複製およびデータ複製の制限についての詳細は、『[SOAP API 開発者ガイド](#)』の「データ複製」を参照してください。

URI

/vXX.X/subjects/**sObjectName**/updated/?start=**startDateAndTime**&end=**endDateAndTime**

形式

JSON、XML

HTTP メソッド

GET

認証

Authorization: Bearer **token**

パラメータ

パラメータ	説明
start	データを取得する期間の開始日時 (ローカル時間ではなく協定世界時 (UTC))。API は、指定された dateTime 値の秒の値を切り捨てます (たとえば、12:30:15 は 12:30:00 UTC となります)。日時は、ISO 8601 形式 (YYYY-MM-DDThh:mm:ss+hh:mm) で指定する必要があります。start の日付/時間値は、end の値より過去の日時でなければなりません。
end	データを取得する期間の終了日時 (ローカル時間ではなく協定世界時 (UTC))。API は、指定された dateTime 値の秒の値を切り捨てます (たとえば、12:35:15 は 12:35:00 UTC となります)。日時は、ISO 8601 形式 (YYYY-MM-DDThh:mm:ss+hh:mm) で指定する必要があります。

応答形式

プロパティ	型	説明
ids	array	要求で指定された開始日と終了日を満たす更新されたレコードの配列。各エントリにはレコード ID が含まれます。
latestDateCovered	String	要求の対象となる最終日の ISO 8601 形式のタイムスタンプ (ローカル時間ではなく協定世界時 (UTC) タイムゾーン)。

例

更新された項目のリストを取得する例は、「[特定の期間に更新されたレコードのリストの取得](#)」(ページ 48)を参照してください。

sObject Rows

指定されたオブジェクト ID に基づいてレコードにアクセスします。レコードを取得、更新、または削除します。このリソースは、項目値の取得にも使用できます。レコードまたは項目を取得するには GET メソッド、レコードを削除するには DELETE メソッド、レコードを更新するには PATCH メソッドを使用します。

新規レコードを作成するには、[sObject Basic Information](#) リソースを使用します。

URI

/vXX.X/subjects/*sObjectName*/*id*/

形式

JSON、XML

HTTP メソッド

GET、PATCH、DELETE

認証

Authorization: Bearer *token*

パラメータ

パラメータ	説明
fields	値を返すために使用される項目のリスト (省略可能)

例

- GET を使用して項目値を取得する例は、「[レコードから項目値を取得する](#)」(ページ 38)を参照してください。
- PATCH を使用してレコードを更新する例は、「[レコードを更新する](#)」(ページ 37)を参照してください。
- DELETE を使用してレコードを削除する例は、「[レコードを削除する](#)」(ページ 38)を参照してください。
- オブジェクトの blob データを更新する例は、「[Blob データを挿入または更新する](#)」(ページ 43)を参照してください。

sObject Rows by External ID

指定された外部 ID 項目の値に基づいて、新しいレコードを作成するか、既存のレコードを更新 (レコードを Upsert) します。

- 指定された値が存在しない場合、新しいレコードが作成されます。
- 指定された値のレコードが存在する場合、リクエストボディに指定された項目値が更新されます。

- 値が一意でない場合、REST API によって、一致するレコードのリストと共に HTTP 状況コード 300 が返されます。



メモ: リクエストボディに ID または外部 ID 項目を指定してはいけません。指定すると、エラーが発生します。

URI

/vXX.X/subjects/*sObjectName*/*fieldName*/*fieldValue*

形式

JSON、XML

HTTP メソッド

HEAD、GET、PATCH、DELETE

認証

Authorization: Bearer *token*

パラメータ

なし

例

- 外部 ID に基づいてレコードを取得する例は、「[外部 ID を使用してレコードを取得する](#)」(ページ 39)を参照してください。
- 外部 ID に基づいてレコードを作成および更新する例は、「[外部 ID を使用してレコードを挿入/更新 \(Upsert\) する](#)」(ページ 39)を参照してください。

sObject Blob Retrieve

個別のレコードから指定された blob 項目を取得します。

URI

/vXX.X/subjects/*sObjectName*/*id*/*blobField*

形式

blob 項目にはバイナリデータが含まれるため、このデータの取得に JSON または XML を使用することはできません。

HTTP メソッド

GET

認証

Authorization: Bearer *token*

パラメータ

不要

例

Attachment または Document から blob データを取得する例は、「[レコードから添付ファイルコンテンツを取得する](#)」(ページ 43)を参照してください。

エラー応答

「[状況コードとエラー応答](#)」(ページ 88)を参照してください。

sObject CompactLayouts

特定のオブジェクトのコンパクトレイアウトのリストを返します。このリソースは REST API バージョン 29.0 以降で使用できます。

構文

URI

特定のオブジェクトのコンパクトレイアウトの説明については、`/vXX.X/subjects/Object/describe/compactLayouts/` を使用します。

形式

JSON、XML

HTTP メソッド

HEAD, GET

認証

Authorization: Bearer **token**

要求パラメータ

不要

例

コンパクトレイアウトの取得

```
curl https://na1.salesforce.com/services/data/v29.0/subjects/Object/describe/compactLayouts/ -H "Authorization: Bearer token"
```

JSON レスポンスボディの例

```
{
  "compactLayouts" : [ ],
  "defaultCompactLayoutId" : "0AH000000000000GAA",
  "recordTypeCompactLayoutMappings" : [ {
    "recordTypeId" : "012000000000000AAA",
    "compactLayoutId" : "Compact Layout ID"
  } ]
}
```

オブジェクトのコンパクトレイアウトを定義していない場合は、compactLayoutId が Null として返されます。

sObject Layouts

パブリッシャーアクションを含む、レイアウトおよび説明のリストを返します。項目のリストおよびレイアウト名が返されます。このリソースは REST API バージョン 28.0 以降で使用できます。パブリッシャーアクションを使用する場合は、「[Quick Actions](#)」も参照してください。



メモ: アプリケーションでは、QuickAction はアクションまたはパブリッシャーアクションと呼ばれます。

URI

グローバルパブリッシャーレイアウトの説明を返すための URI は、`/vXX.X/subjects/Global/describe/layouts/` です。

特定のオブジェクトのレイアウトの説明については、`/vXX.X/subjects/Object/describe/layouts/` を使用します。

形式

JSON、XML

HTTP メソッド

HEAD, GET

認証

Authorization: Bearer **token**

パラメータ

不要

グローバルパブリッシャーレイアウトの取得例

```
curl https://na1.salesforce.com/services/data/v28.0/subjects/Global/describe/layouts/
-H "Authorization: Bearer token"
```

JSON レスポンスボディ `contactlayout.json` ファイルの例

```
[ { "name" : "contactlayout",
  "searchColumns" : [ { "field" : "Account.Name",
    "format" : null,
    "label" : "Account Name",
    "name" : "Name"
  },
  { "field" : "Account.Site",
    "format" : null,
    "label" : "Account Site",
    "name" : "Site"
  },
  { "field" : "Account.Phone",
    "format" : null,
    "label" : "Phone",
    "name" : "Phone"
  },
  { "field" : "User.Alias",
    "format" : null,
```

```

        "label" : "Account Owner Alias",
        "name" : "Owner.Alias"
    }
]
},
{ "label" : "Search Results",
  "limitRows" : 25,
  "searchColumns" : [ { "field" : "Contact.Name",
    "format" : null,
    "label" : "Name",
    "name" : "Name"
  },
    { "field" : "Account.Name",
      "format" : null,
      "label" : "Account Name",
      "name" : "Account.Name"
    },
    { "field" : "Account.Site",
      "format" : null,
      "label" : "Account Site",
      "name" : "Account.Site"
    },
    { "field" : "Contact.Phone",
      "format" : null,
      "label" : "Phone",
      "name" : "Phone"
    },
    { "field" : "Contact.Email",
      "format" : null,
      "label" : "Email",
      "name" : "Email"
    },
    { "field" : "User.Alias",
      "format" : null,
      "label" : "Contact Owner Alias",
      "name" : "Owner.Alias"
    }
  ],
},
{ "label" : "Search Results",
  "limitRows" : 25,
  "searchColumns" : [ { "field" : "Lead.Name",
    "format" : null,
    "label" : "Name",
    "name" : "Name"
  },
    { "field" : "Lead.Title",
      "format" : null,
      "label" : "Title",
      "name" : "Title"
    },
    { "field" : "Lead.Phone",
      "format" : null,
      "label" : "Phone",
      "name" : "Phone"
    },
    { "field" : "Lead.Company",
      "format" : null,
      "label" : "Company",
      "name" : "Company"
    },
    { "field" : "Lead.Email",
      "format" : null,
      "label" : "Email",
      "name" : "Email"
    },
    { "field" : "Lead.Status",
      "format" : null,
      "label" : "Lead Status",

```

```

      "name" : "toLabel (Status) "
    },
    {
      "field" : "Name.Alias",
      "format" : null,
      "label" : "Owner Alias",
      "name" : "Owner.Alias"
    }
  ]
},
]

```

sObject Quick Actions

パブリッシャーアクションおよび詳細のリストを返します。このリソースは REST API バージョン 28.0 以降で使用できます。パブリッシャーアクションを使用する場合は、「[Quick Actions](#)」も参照してください。



メモ: アプリケーションでは、QuickAction はアクションまたはパブリッシャーアクションと呼ばれます。

URI

グローバルアクションだけでなく特定のオブジェクトのアクションを返すには、`/vXX.X/objects/object/quickActions/` を使用します。

特定のアクションを返すには、`/vXX.X/objects/object/quickActions/{アクション名}` を使用します。

特定のアクションの説明の詳細を返すには、`/vXX.X/objects/object/quickActions/{アクション名}/describe/` を使用します。

デフォルトの項目値を含め、特定のアクションのデフォルト値を返すには、`services/data/vXX.X/objects/object/quickActions/{アクション名}/defaultValues/` を使用します。

API バージョン 28.0 で、アクションのデフォルト値を評価するには、`vXX.X/objects/object/quickActions/{アクション名}/defaultValues/{親 ID}` を使用します。

API バージョン 29.0 以降で、アクションのデフォルト値を評価するには、`vXX.X/objects/object/quickActions/{アクション名}/defaultValues/{コンテキスト ID}` を使用します。

これは、{コンテキスト ID} オブジェクトに固有のデフォルト値を返します。

形式

JSON、XML

HTTP メソッド

HEAD、GET、POST

認証

Authorization: Bearer **token**

パラメータ

不要

取引先アクションを取得する場合の例

```
curl https://na1.salesforce.com/services/data/v28.0/subjects/Account/quickActions -H "Authorization: Bearer token"
```

アクションを使用して取引先で取引先責任者を作成する場合の例

```
curl https://na1.salesforce.com/services/data/v28.0/subjects/Account/quickActions/CreateContact -H 'Authorization: Bearer access_token -H "Content-Type: application/json" -d @newcontact.json'
```

JSON リクエストボディ **newcontact.json** ファイルの例

```
{
  "contextId" : "001D000000JRSGf",
  "record" : { "LastName" : "Smith" }
}
```

考慮事項

- リソースからは、要求したアクションに加え、すべてのアクション(グローバルおよび標準)が返されます。

sObject User Password

ユーザパスワードを設定またはリセットしたり、ユーザパスワードに関する情報を取得したりします。このリソースは REST API バージョン 24.0 以降で使用できます。

URI

/vXX.X/subjects/User/**ユーザ ID**/password

セルフサービスユーザのパスワードを管理する場合、URI は次のとおりです。

/vXX.X/subjects/SelfServiceUser/**セルフサービスユーザ ID**/password

形式

JSON、XML

HTTP メソッド

HEAD、GET、POST、DELETE

認証

Authorization: Bearer **token**

パラメータ

不要

例

パスワード情報の取得、パスワードの設定、パスワードのリセットの例は、「[ユーザパスワードを管理する](#)」(ページ 56)を参照してください。

考慮事項

- セッションにユーザ情報にアクセスする権限がない場合、INSUFFICIENT_ACCESS エラーが返されます。
- このリソースを使用して新しいパスワードを設定する場合、新しいパスワードは、組織のパスワードポリシーに適合している必要があり、適合していない場合は、INVALID_NEW_PASSWORD エラー応答が返されます。
- 1つの要求で設定可能なパスワードは1つのみです。
- このリソースの DELETE メソッドを使用する場合、Salesforce は、ユーザパスワードを自動生成されたパスワードにリセットし、応答で返します。

AppMenu

Salesforce アプリケーションドロップダウンメニューまたは Salesforce1 ナビゲーションメニューの項目のリストを返します。

構文

URI

Salesforce アプリケーションドロップダウンメニュー項目のリストを返すには、URI に `/vXX.X/appMenu/AppSwitcher/` を使用します。

Salesforce1 ナビゲーションメニュー項目のリストを返すには、URI に `/vXX.X/appMenu/Salesforce1/` を使用します。

適用開始バージョン

29.0

形式

JSON、XML

HTTP メソッド

GET、HEAD

認証

Authorization: Bearer **token**

リクエストボディ

None

要求パラメータ

不要

例

appMenu の種別の取得

```
curl https://na1.salesforce.com/services/data/v29.0/appMenu/ -H "Authorization: Bearer token"
```

/vXX.X/appMenu/AppSwitcher/ のレスポンスボディの例

```
{
  "appMenuItems" : [ {
    "type" : "Tabset",
    "content" : null,
    "icons" : null,
    "colors" : null,
    "label" : "Sales",
    "url" : "/home/home.jsp?tsid=02uxx00000056Sq"
  }, {
    "type" : "Tabset",
    "content" : null,
    "icons" : null,
    "colors" : null,
    "label" : "Call Center",
    "url" : "/home/home.jsp?tsid=02uxx00000056Sr"
  }, {
    "type" : "Tabset",
    "content" : null,
    "icons" : null,
    "colors" : null,
    "label" : "Marketing",
    "url" : "/home/home.jsp?tsid=02uxx00000056St"
  }, {
    "type" : "Tabset",
    "content" : null,
    "icons" : null,
    "colors" : null,
    "label" : "Salesforce Chatter",
    "url" : "/home/home.jsp?tsid=02uxx00000056Su"
  }, {
    "type" : "Tabset",
    "content" : null,
    "icons" : null,
    "colors" : null,
    "label" : "Community",
    "url" : "/home/home.jsp?tsid=02uxx00000056Sw"
  }, {
    "type" : "Tabset",
    "content" : null,
    "icons" : null,
    "colors" : null,
    "label" : "App Launcher",
    "url" : "/app/mgmt/applauncher/appLauncher.apexp?tsid=02uxx00000056Sx"
  } ]
}
```

/vXX.X/appMenu/Salesforce1/ のレスポンスボディの例

```
{
  "appMenuItems" : [ {
    "type" : "Standard.Search",
    "content" : null,
    "icons" : null,
    "colors" : null,
```

```

    "label" : "Smart Search Items",
    "url" : "/search"
  }, {
    "type" : "Standard.MyDay",
    "content" : null,
    "icons" : null,
    "colors" : null,
    "label" : "Today",
    "url" : "/myDay"
  }, {
    "type" : "Standard.Tasks",
    "content" : null,
    "icons" : null,
    "colors" : null,
    "label" : "Tasks",
    "url" : "/tasks"
  }, {
    "type" : "Standard.Dashboards",
    "content" : null,
    "icons" : null,
    "colors" : null,
    "label" : "Dashboards",
    "url" : "/dashboards"
  }, {
    "type" : "Tab.flexiPage",
    "content" : "MySampleFlexiPage",
    "icons" : [ {
      "contentType" : "image/png",
      "width" : 32,
      "height" : 32,
      "theme" : "theme3",
      "url" : "http://myorg.com/img/icon/custom51_100/bell132.png"
    }, {
      "contentType" : "image/png",
      "width" : 16,
      "height" : 16,
      "theme" : "theme3",
      "url" : "http://myorg.com/img/icon/custom51_100/bell116.png"
    }, {
      "contentType" : "image/svg+xml",
      "width" : 0,
      "height" : 0,
      "theme" : "theme4",
      "url" : "http://myorg.com/img/icon/t4/custom/custom53.svg"
    }, {
      "contentType" : "image/png",
      "width" : 60,
      "height" : 60,
      "theme" : "theme4",
      "url" : "http://myorg.com/img/icon/t4/custom/custom53_60.png"
    }, {
      "contentType" : "image/png",
      "width" : 120,
      "height" : 120,
      "theme" : "theme4",
      "url" : "http://myorg.com/img/icon/t4/custom/custom53_120.png"
    } ],
    "colors" : [ {
      "context" : "primary",
      "color" : "FC4F59",
      "theme" : "theme4"
    }, {
      "context" : "primary",
      "color" : "FC4F59",
      "theme" : "theme3"
    } ],
    "label" : "My App Home Page",
    "url" : "/servlet/servlet.Integration?lid=01rxx0000000Vsd&ic=1"
  }, {

```

```

"type" : "Tab.apexPage",
"content" : "/apex/myapexpage",
"icons" : [ {
  "contentType" : "image/png",
  "width" : 32,
  "height" : 32,
  "theme" : "theme3",
  "url" : "http://myorg.com/img/icon/cash32.png"
}, {
  "contentType" : "image/png",
  "width" : 16,
  "height" : 16,
  "theme" : "theme3",
  "url" : "http://myorg.com/img/icon/cash16.png"
}, {
  "contentType" : "image/svg+xml",
  "width" : 0,
  "height" : 0,
  "theme" : "theme4",
  "url" : "http://myorg.com/img/icon/t4/custom/custom41.svg"
}, {
  "contentType" : "image/png",
  "width" : 60,
  "height" : 60,
  "theme" : "theme4",
  "url" : "http://myorg.com/img/icon/t4/custom/custom41_60.png"
}, {
  "contentType" : "image/png",
  "width" : 120,
  "height" : 120,
  "theme" : "theme4",
  "url" : "http://myorg.com/img/icon/t4/custom/custom41_120.png"
} ],
"colors" : [ {
  "context" : "primary",
  "color" : "3D8D8D",
  "theme" : "theme4"
}, {
  "context" : "primary",
  "color" : "3D8D8D",
  "theme" : "theme3"
} ],
"label" : "label",
"url" : "/servlet/servlet.Integration?lid=01rxx0000000VyB&ic=1"
} ]
}

```

FlexiPage

フレキシブルページとその詳細のリストを返します。返される情報には、フレキシブルページの範囲、各範囲内のコンポーネント、各コンポーネントのプロパティ、および関連付けられた QuickActions が含まれます。このリソースは API バージョン 29.0 以降で使用できます。

構文

URI

フレキシブルページの詳細をすべて返すには、`/vXX.X/flexiPage/フレキシブルページの ID`を使用します。

形式

JSON、XML

HTTP メソッド

HEAD, GET

認証

Authorization: Bearer **token**

パラメータ

不要

例

ルートフレキシブルページリソースの取得

```
curl https://na1.salesforce.com/services/data/v29.0/flexiPage/ -H "Authorization: Bearer token"
```

ID が 0M0xx0000000001CAA であるフレキシブルページの取得

```
curl https://na1.salesforce.com/services/data/v29.0/flexiPage/0M0xx0000000001CAA -H "Authorization: Bearer token"
```

/vXX.X/flexiPage/ のリクエストボディの例

不要

/vXX.X/flexiPage/ のレスポンスボディの例

```
{
  "urls" : {
    "flexiPage" : "/services/data/v29.0/flexiPage",
    "rowTemplate" : "/services/data/v29.0/flexiPage/{ID}"
  }
}
```

/vXX.X/flexiPage/{フレキシブルページの ID} のリクエストボディの例

不要

/vXX.X/flexiPage/{フレキシブルページの ID} のレスポンスボディの例



メモ: このコード例には、quickActionList 情報が含まれています。REST API でのクイックアクション (別名パブリッシャーアクション) についての詳細は、「[Quick Actions](#)」および「[sObject Quick Actions](#)」を参照してください。

```
{
  "name": "DeveloperNameOfFlexiPage",
  "id": "0M0xx0000000001CAA",
  "label": "FlexiPage Label",
  "quickActionList": {
    "quickActionListItems": [
      {
        "type": "Post",
        "label": "Post",
        "quickActionName": "FeedItem.TextPost",
        "targetSobjectType": null,

```

```

        "iconUrl": null,
        "miniIconUrl": null
      },
      {
        "type": "Create",
        "label": "testFlexiQuickAction",
        "quickActionName": "flexiAction",
        "targetSobjectType": "Contact",
        "iconUrl": "http://{SALESFORCE-APPSERVER-DOMAIN}/img/icon/contacts32.png",
        "miniIconUrl":
"http://{SALESFORCE-APPSERVER-DOMAIN}/img/icon/contacts16.png"
      }
    ],
    "regions": [
      {
        "name": "main",
        "components": [
          {
            "properties": [
              {
                "name": "entityName",
                "value": "Account"
              },
              {
                "name": "filterName",
                "value": "MyAccounts"
              }
            ],
            "typeName": "filterList",
            "typeNamespace": "force"
          }
        ]
      }
    ]
  }
}

```

このサンプルコードの内容は次のとおりです。

- name — 範囲の名前
- components — 範囲の Aura コンポーネントの配列
- properties — コンポーネントのプロパティの配列
- typeName — Aura コンポーネントの名前
- typeNamespace — Aura コンポーネントの名前空間

Query

指定された SOQL クエリを実行します。

クエリ結果が大きすぎる場合、応答には、結果の最初のバッチと応答の `nextRecordsUrl` 項目のクエリ識別子が含まれます。識別子は、追加の要求で次のバッチを取得するために使用できます。

URI

`/vXX.X/query/?q=SOQL クエリ`

最初の結果が大きすぎる場合、以降のクエリ結果は次のように取得します。

`/vXX.X/query/クエリの識別子`

形式

JSON、XML

HTTP メソッド

GET

認証Authorization: Bearer **token****パラメータ**

パラメータ	説明
q	SOQL クエリ。有効な URI を作成するには、クエリ文字列内のスペースを「+」文字で置き換える必要があります。たとえば、クエリパラメータ文字列は「SELECT+Name+FROM+MyObject」のようになります。

例

クエリの作成と、クエリ識別子を使用した以降のクエリ結果の取得の例は、「[SOQL クエリを実行する](#)」(ページ 49)を参照してください。

SOQL についての詳細は、『[Force.com SOQL and SOSL Reference](#)』を参照してください。クエリのバッチサイズについての詳細は、『[SOAP API 開発者ガイド](#)』の「[クエリのバッチサイズの変更](#)」を参照してください。

QueryAll

指定された SOQL クエリを実行します。Query リソースとは異なり、QueryAll は merge または delete によって削除されたレコードを返します。また、QueryAll はアーカイブ済みの ToDo と行動のレコードの情報を返します。QueryAll は、API バージョン 29.0 以降で使用できます。

クエリ結果が大きすぎる場合、応答には、結果の最初のバッチと応答の nextRecordsUrl 項目のクエリ識別子が含まれます。識別子は、追加の要求で次のバッチを取得するために使用できます。nextRecordsUrl の URL に query が指定されている場合でも、最初の QueryAll 要求の残りの結果が提供されます。残りの結果には、最初のクエリに一致した削除されたレコードが含まれます。

URI/vXX.X/queryAll/?q=**SOQL クエリ**

最初の結果が大きすぎる場合、以降のクエリ結果は次のように取得します。

/vXX.X/queryAll/**クエリの識別子****形式**

JSON、XML

HTTP メソッド

GET

認証

Authorization: Bearer **token**

パラメータ

パラメータ	説明
q	SOQL クエリ。有効な URI を作成するには、クエリ文字列内のスペースを「+」文字で置き換える必要があります。たとえば、クエリパラメータ文字列は「SELECT+Name+FROM+MyObject」のようになります。

例

- 削除された項目を含むクエリを作成する例は、「[削除された項目を含む SOQL クエリを実行する](#)」(ページ 50)を参照してください。
- クエリ識別子を使用して追加の結果を取得するクエリの例は、「[SOQL クエリの残りの結果の取得](#)」(ページ 51)を参照してください。

SOQL についての詳細は、『[Force.com SOQL and SOSL Reference](#)』を参照してください。クエリのバッチサイズについての詳細は、『[SOAP API 開発者ガイド](#)』の「[クエリのバッチサイズの変更](#)」を参照してください。

Quick Actions

グローバルパブリッシャーアクションおよび標準アクションのリストを返します。このリソースは REST API バージョン 28.0 以降で使用できます。パブリッシャーアクションを使用する場合は、「[sObject Quick Actions](#)」も参照してください。



メモ: アプリケーションでは、QuickAction はアクションまたはパブリッシャーアクションと呼ばれます。

URI

/vXX.X/quickActions/

形式

JSON、XML

HTTP メソッド

HEAD、GET、POST

認証

Authorization: Bearer **token**

パラメータ

不要

グローバルクイックアクションを取得する場合の使用例

```
curl https://na1.salesforce.com/services/data/v28.0/quickActions/ -H "Authorization: Bearer token"
```

アクションを使用して取引先責任者を作成する場合の例

```
curl https://na1.salesforce.com/services/data/v28.0/quickActions/CreateContact -H 'Authorization: Bearer access_token -H "Content-Type: application/json" -d @newcontact.json'
```

JSON リクエストボディ newcontact.json ファイルの例

```
{
  "record" : { "LastName" : "Smith" }
}
```

Search

指定された SOSL 検索を実行します。検索文字列は URL 符号化されている必要があります。

URI

/vXX.X/search/?q=**SOSL 検索文字列**

形式

JSON、XML

HTTP メソッド

GET

認証

Authorization: Bearer **token**

パラメータ

パラメータ	説明
q	適切に URL 符号化された SOSL ステートメント。

例

「[文字列を検索する](#)」(ページ 51)を参照してください。

SOSL についての詳細は、[『Force.com SOQL and SOSL Reference』](#)を参照してください。

Search Scope and Order

ログインユーザのデフォルトのグローバル検索範囲内にあるオブジェクトの順序付きリストを返します。グローバル検索は、操作するオブジェクトとそれら进行操作する頻度を追跡し、それに基づいて検索結果を編成します。最もよく使用されるオブジェクトは、リストの最上部に表示されます。

返されるリストには、ユーザの検索結果ページの固定表示オブジェクトを含め、ユーザのデフォルトの検索範囲でのオブジェクト順が反映されます。このコールは、最適化されたグローバル検索範囲を使用してカスタム検索結果ページを実装する場合に役立ちます。検索文字列は URL 符号化されている必要があります。

URI

`/vXX.X/search/scopeOrder`

形式

JSON、XML

HTTP メソッド

GET

認証

Authorization: Bearer **token**

例

「[デフォルトの検索範囲と検索順序の取得](#)」を参照してください。

Search Result Layouts

クエリ文字列に含まれるオブジェクトの検索結果レイアウトに関する情報を返します。このコールでは、検索結果ページに列として表示される項目のリスト、最初のページに表示される行数、および検索結果ページで使われるラベルがオブジェクトごとに返されます。このコールでは、1回のクエリで100個までのオブジェクトの一括取得をサポートしています。

URI

`/vXX.X/search/layout/?q=カンマで区切られたオブジェクトのリスト`

形式

JSON、XML

HTTP メソッド

GET

認証

Authorization: Bearer **token**

例

[オブジェクトの検索結果レイアウトの取得](#)

Recently Viewed Items

現在のユーザが表示または参照した、最近参照された項目を取得します。Salesforce では、レコード参照に関する情報がインターフェースに保存され、その情報を使用して、サイドバーや検索のオートコンプリートオプションなどで、最近表示および参照したレコードのリストが生成されます。

このリソースは、最近使ったデータの情報のみにアクセスします。最近参照したデータのリストを変更するには、SOQL [クエリ](#) で FOR VIEW 句または FOR REFERENCE 句を指定して、最近参照した情報を直接更新する必要があります。

URI

/vXX.X/recent

形式

JSON、XML

HTTP メソッド

GET

認証

Authorization: Bearer **token**

パラメータ

パラメータ	説明
limit	返されるレコードの最大数を指定するパラメータ (省略可能)。このパラメータが指定されていない場合、返されるレコードのデフォルトの最大数は RecentlyViewed のエントリの最大数 (オブジェクトあたり 200 レコード) になります。

例

- 最近参照した項目のリストを取得する例については、「[最近参照したレコードの表示](#)」(ページ 55)を参照してください。
- レコードを最近参照したデータとして設定する例は、「[最近参照したデータとしてレコードをマーク](#)」(ページ 55)を参照してください。

Themes

Salesforce アプリケーションのテーマで使用するアイコンと色のリストを取得します。テーマ情報は、Salesforce UI のアイコンと色を使用する組織内のオブジェクトに提供されます。

構文

URI

/vXX.X/theme

適用開始バージョン

29.0

形式

JSON、XML

HTTP メソッド

GET

認証

Authorization: Bearer **token**

リクエストボディ

None

要求パラメータ

なし

応答データ

テーマ項目の配列。各テーマ項目には次の項目が含まれます。

名前	型	説明
colors	array	テーマの色の配列。
icons	array	テーマアイコンの配列。
name	string	テーマの色とアイコンが関連付けられるオブジェクトの名前。

各テーマの色には次の項目が含まれます。

名前	型	説明
color	string	Web 色の RGB 形式で示される色 (00FF00 など)。
context	string	オブジェクトでその色がメインの色 (primary) であるかどうかを決定する色のコンテキスト。
theme	string	<p>関連付けられたテーマ。値には、次のものがあります。</p> <ul style="list-style-type: none"> • theme2: Spring '10 より前に使用されていた Salesforce テーマ • theme3: Spring '10 で導入された現在の Salesforce テーマ • theme4: Winter '14 で導入されたモバイルタッチスクリーンバージョンの Salesforce テーマ • custom: カスタムアイコンに関連付けられたテーマ

各テーマアイコンには次の項目が含まれます。

名前	型	説明
contentType	string	アイコンのコンテンツタイプは、「image/png」などです。
height	number	アイコンの高さ (ピクセル単位)。アイコンのコンテンツタイプが SVG タイプである場合、高さ と 幅 の値は使用されません。

名前	型	説明
theme	string	<p>関連付けられたテーマ。値には、次のものがあります。</p> <ul style="list-style-type: none"> • theme2: Spring '10 より前に使用されていた Salesforce テーマ • theme3: Spring '10 で導入された現在の Salesforce テーマ • theme4: Winter '14 で導入されたモバイルタッチスクリーンバージョンの Salesforce テーマ • custom: カスタムアイコンに関連付けられたテーマ
url	string	このアイコンの完全修飾 URL。
width	number	アイコンの幅 (ピクセル単位)。アイコンのコンテンツタイプが SVG タイプである場合、高さとの値は使用されません。

例

services/data/v29.0/theme の要求を使用した JSON 応答の例を次に示します。

```
{
  "themeItems" : [
    {
      "name" : "Merchandise__c",
      "icons" : [
        {
          "contentType" : "image/png",
          "width" : 32,
          "url" : "https://na1.salesforce.com/img/icon/computer32.png",
          "height" : 32,
          "theme" : "theme3"
        },
        {
          "contentType" : "image/png",
          "width" : 16,
          "url" : "https://na1.salesforce.com/img/icon/computer16.png",
          "height" : 16,
          "theme" : "theme3"
        }
      ],
      "colors" : [
        {
          "context" : "primary",
          "color" : "6666CC",
          "theme" : "theme3"
        },
        {
          "context" : "primary",
          "color" : "66895F",
          "theme" : "theme4"
        }
      ],
      ...
    }
  ],
  ...
}
```

ヘッダー

このセクションでは、REST API に使用されるカスタム HTTP の要求ヘッダーと応答ヘッダーを示します。

- [割り当てルールヘッダー](#)

割り当てルールヘッダー

割り当てルールヘッダーは、ケースまたはリードの作成時または更新時に適用される要求ヘッダーです。有効化されていると、有効な割り当てルールが使用されます。無効化されていると、有効な割り当てルールは適用されません。有効な AssignmentRule ID が指定されていると、AssignmentRule が適用されます。要求にヘッダーが指定されていないと、REST API のデフォルトにより有効な割り当てルールが使用されます。

ヘッダーの項目名と値

項目名

SForce-Auto-Assign

項目値

- TRUE。作成済みまたは更新済みのケースまたはリードに、有効な割り当てルールが適用されます。これは、要求にヘッダーが指定されていない場合のデフォルトです。
- FALSE。作成済みまたは更新済みのケースまたはリードに、有効な割り当てルールは適用されません。
- 有効な AssignmentRule ID。作成済みのケースまたはリードに、指定の AssignmentRule が適用されます。

TRUE または FALSE の値では、大文字と小文字は区別されません。

例

SForce-Auto-Assign: FALSE

Limit Info ヘッダー

ヘッダーの項目名と値

Limit Info ヘッダーは、REST API への各コールで返される応答ヘッダーです。このヘッダーは、組織の制限情報を返します。このヘッダーを使用して、組織にコールを行うときの API 制限を監視します。

項目名

Sforce-Limit-Info

項目値

- api-usage — コールが行われた組織の API 使用状況を nn/nnnn 形式で指定します。最初の数値は使用された API コール数で、2 番目の数値は組織の API 制限数です。

例

Sforce-Limit-Info: api-usage=14/5000

次に、Merchandise レコードの REST 要求に対する応答の例を示します。

```
HTTP/1.1 200 OK
Date: Mon, 20 May 2013 22:21:46 GMT
Sforce-Limit-Info: api-usage=18/5000
Last-Modified: Mon, 20 May 2013 20:49:32 GMT
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
{
```

```

"attributes" : {
  "type" : "Merchandise__c",
  "url" : "/services/data/v29.0/subjects/Merchandise__c/a00D00000008pQSNIA2"
},
"Id" : "a00D00000008pQSNIA2",
"OwnerId" : "005D00000001QX8WIAW",
"IsDeleted" : false,
"Name" : "Phone Case - iPhone 4/4S",
"CreatedDate" : "2013-05-20T20:49:32.000+0000",
"CreatedById" : "005D00000001QX8WIAW",
"LastModifiedDate" : "2013-05-20T20:49:32.000+0000",
"LastModifiedById" : "005D00000001QX8WIAW",
"SystemModstamp" : "2013-05-20T20:49:32.000+0000",
"LastActivityDate" : null,
"LastViewedDate" : "2013-05-20T22:19:56.000+0000",
"LastReferencedDate" : "2013-05-20T22:19:56.000+0000",
"Description__c" : "Phone Case for iPhone 4/4S",
"Price__c" : 16.99,
"Stock_Price__c" : 12.99,
"Total_Inventory__c" : 108.0
}

```

状況コードとエラー応答

エラーが発生した場合、または応答が正常な場合のどちらでも、応答ヘッダーには HTTP コードが含まれ、レスポンスボディには通常、次の情報が含まれます。

- HTTP 応答コード
- HTTP 応答コードに付随するメッセージ
- エラーが発生した項目またはオブジェクト (応答がエラーに関する情報を返す場合)

HTTP 応答コード 説明	
200	GET または HEAD 要求の「OK」成功コードです。
201	POST 要求の「Created」成功コードです。
204	DELETE 要求の「No Content」成功コードです。
300	外部IDが複数のレコードに存在する場合に返される値です。レスポンスボディには、一致するレコードのリストが含まれます。
304	要求のコンテンツが、指定された日時から変更されていません。日時は If-Modified-Since ヘッダーで指定されます。例については、 「オブジェクトのメタデータの変更の取得」 を参照してください。
400	要求が実行されませんでした。通常、JSON または XML のボディに含まれるエラーが原因です。
401	使用されたセッションIDまたは OAuth トークンが期限切れが無効です。レスポンスボディに message および errorCode が含まれます。
403	要求が却下されました。ログインユーザーに適切な権限があることを確認してください。
404	要求されたリソースが見つかりませんでした。URI にエラーがないか確認し、共有の問題がないことを確認してください。

HTTP 応答コード	説明
405	Request-Line に指定されたメソッドは、URI に指定されたリソースには許可されていません。
415	要求内のエンティティは、指定されたメソッドではサポートされていない形式です。
500	Force.com 内でエラーが発生したため、要求を完了できませんでした。salesforce.com カスタマーサポートにお問い合わせください。

例: ID が不正

JSON または XML (`request_body.json` または `request_body.xml`) を使用する要求に存在しない ID を使用した場合。

```
{
  "fields" : [ "Id" ],
  "message" : "Account ID: id value of incorrect type: 001900K0001pPuOAAU",
  "errorCode" : "MALFORMED_ID"
}
```

例: リソースが存在しない

存在しないリソースを要求した場合。たとえば、誤ったスペルのオブジェクト名を使用してレコードの作成を試みた場合など。

```
{
  "message" : "The requested resource does not exist",
  "errorCode" : "NOT_FOUND"
}
```