

松 山 大 学 論 集
第 32 卷 第 2 号 抜 刷
2 0 2 0 年 6 月 発 行

EDINET API からの情報収集

中 溝 晃 介

EDINET API からの情報収集

中 溝 晃 介

第1節 はじめに

AI（人工知能）の普及に伴い、世間では人の職が奪われるとも言われている。データサイエンスもまた注目を浴びている用語のひとつである。膨大なデータから何らかの知識（関係性）を導くものである。

データを分析するためには、当然ながら分析する対象のデータが必要になる。そのデータを集めるという作業から始めなくてはならない。データを収集するという仕事がある。例えば、図書館や資料室に行き、該当する資料やデータをかき集めることである。資料は紙であることが多く、その内容をデータとしてパソコンに入力することもしばしば起こる。

会計書類である有価証券報告書も以前は紙媒体であった。1部上場会社であれば1冊2,550円（税抜き）であり、1年間分のセット価格ともなれば180万円を超える。この金額を出しても、データは紙である。

今日では、金融庁のEDINETを通じて有価証券報告書を閲覧することができる。1社でも、1年分でも利用する費用はかからない（通信費は別途必要）。一方で、1年分の有価証券報告書が4,000社であったとして、4,000社分の有価証券報告書を取得することは容易ではない。現在のEDINETには、2019年1月1日から2019年12月31日までの有価証券報告書を一括でダウンロードする、といった機能はない。1社ずつ手作業で行うか、プログラムを作成し（一部）自動化させるかのどちらかだ。

このような中、金融庁はEDINET APIを2019年3月に公開した。APIの公

開は、プログラミングを行う側にとって、作業効率を改善するメリットがある。本稿では、この EDINET API について、どのように利用するのかを説明していく。

第2節 EDINET API

中溝（2018）では、EDINET から有価証券報告書の XBRL データをダウンロードする Python プログラムの作成を行った。Python のライブラリは利用したが、EDINET API が公開されていなかったため、JavaScript を解釈できるクローラー Selenium を用いて自動でダウンロードする作業を実現させた。ここで作成したプログラムでは、XBRL データだけでなく、提出日時や提出書類名といった他の情報も取得できるようになっている。API を利用することで、これらの情報を得る敷居が低くなる。

EDINET API は、利用者が EDINET の画面からではなく、プログラムを介して EDINET のデータベースから効率的にデータを取得できる API（Application Programming Interface）である。EDINET API により、EDINET 利用者は効率的に開示情報を取得することが可能になる¹⁾

EDINET は、2つの API を提供する。1つ目が、提出された書類を把握するための API である。EDINET に提出された書類の一覧を取得する API（以下、書類一覧 API）であり、日付ごとに提出書類情報、書類情報修正情報等を把握することが可能である。書類一覧 API では、「メタデータのみ」または「提供書類一覧及びメタデータ」を取得することができる。ここに、メタデータとは、日付ごとの提出書類一覧に関する基礎情報のことで、件数、更新日時等を含んでいる²⁾

2つ目が、提供された書類を取得するための API である。EDINET に提出された書類を取得する API（以下、書類取得 API）であり、リクエストパラメー

1) 金融庁 [2019], 2頁。

2) 金融庁 [2019], 3頁。

タにより取得する書類の種類を指定することが可能になる³⁾

第3節 書類一覧 API

書類一覧 API は、「メタデータのみ」または「提供書類一覧及びメタデータ」を取得する。API を利用するにあたり、リクエストとレスポンスと呼ばれる概念の理解が重要になる。リクエストが情報を取得する手段に関するもので、レスポンスが取得した情報に関するものである。

第1項 リクエスト

EDINET API はインターネットに接続された機器があれば容易に体験が可能である。次に示す URL を Web ブラウザに入力し、出力された画面が API の結果である。

コード1

```
https://disclosure.edinet-fsa.go.jp/api/v1/documents.json  
?date=2020-04-01
```

※途中で改行しているが、1行一続きの URL である。

出力結果1

```
{  
  "metadata":  
    {  
      "title": "提出された書類を把握するための API",  
      "parameter":  
        {  
          "date": "2020-04-01",  
          "type": "1"  
        },  
    },  
}
```

3) 金融庁 [2019], 3頁。

```
"resultset":
  {
    "count":255
  },
  "processDateTime":"2020-04-15 00:03",
  "status":"200",
  "message":"OK"
}
```

結果の詳細は後述するが、EDINET API は特別なプログラムを使わなくとも、web ブラウザで動作を確認することができる。

具体的にコードの中身を見ていこう。コード1の1行目はエンドポイントといい、EDINET API で事前に設定されている URI⁴⁾ が存在する。その内容は次の通りである。

```
https://disclosure.edinet-fsa.go.jp/api/バージョン/documents.json
```

バージョンは「v+整数」で表記する。2020年4月時点ではバージョンは1であるため、「v1」を用いる。よって、本稿で使用するエンドポイントは次のようになる。

```
https://disclosure.edinet-fsa.go.jp/api/v1/documents.json
```

APIによっては、HTTP メソッドと呼ばれるリソースの取得といった動作を指定するが、EDINET API は GET メソッドを使用するように求められている。GET や POST などのメソッド名を URI に含めることは行わない。

4) URI とは一般的に URL や URN の総称として用いられる。本稿の場合、Web ページのアドレス表記などの URL と同じイメージと捉えても問題はない。

エンドポイントの後ろに、リクエストパラメータを指定することで書類一覧 API のリクエスト URL が完成する。このパラメータは EDINET 側で動くプログラムに情報を送るために、エンドポイントの末尾に付ける変数を指し、「? (クエスチョンマーク)」と「パラメータ名=設定値」を組み合わせる。

パラメータは【表 1】の通りである。

表 1 書類一覧 API のリクエストパラメータ

パラメータ名	項目名	必須	設定値	説明
date	ファイル日付	○	日付 (YYYY-MM-DD 形式)	出力対象とする提出書類一覧のファイル日付 ⁵⁾ を指定する。
type	取得情報	-	1	メタデータのみを取得する。 指定がない場合のデフォルト値である。
			2	提出書類一覧及びメタデータを取得する。

(出所：金融庁 [2019], 7 頁。一部筆者修正)

コード 1 ではパラメータ名「date」を使用し、設定値を「2020-04-01」とした。これを「?」と組み合わせると、「?date=2020-04-01」というリクエストパラメータが完成する。「type」パラメータの設定値は「1」と「2」のどちらかを設定することになる。設定値「1」はデフォルト値とされているため、設定しても「date」のみを設定した場合と結果は同じである。したがって、メタデータのみを取得したい場合とメタデータと提出書類一覧を取得したい場合の 2 種類のリクエスト URL が考えられる。

5) 「ファイル日付」は当日及び土日祝日を含む過去 5 年の日付を指定することができる。これは EDINET の書類保管期限が 5 年と定められているためである。

コード1（再掲）：メタデータのみを取得

```
https://disclosure.edinet-fsa.go.jp/api/v1/documents.json  
?date=2020-04-01
```

コード2：メタデータと提出書類一覧を取得

```
https://disclosure.edinet-fsa.go.jp/api/v1/documents.json  
?date=2020-04-01&type=2
```

第2項 レスポンス

コード1やコード2をリクエストした結果についてみていく。コード1から得られる結果は出力結果1に示した通りである。コード2から得られる出力結果を次に示す。

出力結果2

```
{  
  "metadata":  
  {  
    "title": "提出された書類を把握するためのAPI",  
    "parameter":  
    {  
      "date": "2020-04-01",  
      "type": "2"  
    },  
    "resultset":  
    {  
      "count": 255  
    },  
    "processDateTime": "2020-04-16 00:02",  
    "status": "200",  
    "message": "OK"  
  },  
},
```



```
"results": [
  {
    "seqNumber": 1,
    "docID": "S100IBHG",
    "edinetCode": "E09666",
    "secCode": null,
    "JCN": "9010001062807",
    "filerName": "アムンティ・ジャパン株式会社",
    "fundCode": "G11457",
    "ordinanceCode": "030",
    "formCode": "04A001",
    "docTypeCode": "040",
    "periodStart": null,
    "periodEnd": null,
    "submitDateTime": "2020-04-01 09:00",
    "docDescription": "訂正有価証券届出書
      (内国投資信託受益証券)",
    "issuerEdinetCode": null,
    "subjectEdinetCode": null,
    "subsidiaryEdinetCode": null,
    "currentReportReason": null,
    "parentDocID": "S100H7RN",
    "opeDateTime": null,
    "withdrawalStatus": "0",
    "docInfoEditStatus": "0",
    "disclosureStatus": "0",
    "xbrlFlag": "1",
    "pdfFlag": "1",
    "attachDocFlag": "1",
    "englishDocFlag": "0"
  },

```

②

※以下、別の書類に関する②に含まれる情報が続く。

出力結果2について、①に含まれる情報はコード1から得られる情報と同じである。コード2を入力した場合、①に加えて②の情報が追加されることになる。ひとつの書類に関する情報が②である。別の提出書類があれば、②の情報が下に続く形で表示される。それぞれの情報について、主要な項目について解説していく。出力内容を【表2】に示す。

表2 出力データの内容

No.	項目名	項目ID	説明
1	メタデータ	metadata	メタデータの識別子
2	タイトル	title	APIの名称を出力
3	パラメータ	parameter	リクエストパラメータの識別子
4	ファイル日付	date	指定したファイル日付を出力
5	取得情報	type	指定した取得情報を出力
6	結果セット	resultset	結果セットの識別子
7	件数	count	指定したファイル日付における提出書類一覧の件数を出力
8	書類一覧更新日時	processDateTime	提出書類一覧の更新時間を出力
9	ステータス	status	ステータスコードに記載されたステータスを出力
10	メッセージ	message	ステータスコードに記載されたメッセージを出力
11	提出書類一覧	results	提出書類一覧の識別子
12	連番	seqNumber	ファイル日付ごとの連番
13	書類管理番号	docID	書類管理番号を出力
14	提供者 EDINET コード	edinetCode	提供者の EDINET コードを出力
15	提出者証券コード	secCode	提出者の証券コードを出力
16	提出者法人番号	filerName	提出者の法人番号を出力
17	提出者名	JCN	提出者の名前を出力
18	ファンドコード	fundCode	ファンドコードを出力
19	府令コード	ordinanceCode	府令コードを出力
20	様式コード	formCode	様式コードを出力

21	書類識別コード	docTypeCode	書類識別コードを出力
22	期間（自）	periodStart	期間（自）を出力
23	期間（至）	periodEnd	期間（至）を出力
24	提出日時	submitDateTime	提出日時を出力
25	提出書類概要	docDescription	EDINET の閲覧サイトの書類検索結果画面において、「提出書類」欄に表示される文字列を出力
26	発行会社 EDINET コード	issuerEdinetCode	大量保有について発行会社の EDINET コードを出力
27	対象 EDINET コード	subjectEdinetCode	公開買付けについて対象となる EDINET コードを出力
28	子会社 EDINET コード	subsidiaryEdinetCode	子会社の EDINET コードを出力
29	臨報提出事由	currentReportReason	臨時報告書の提出事由を出力
30	親書類管理番号	parentDocID	親書類管理番号を出力
31	操作日時	opeDateTime	磁気ディスク提出及び紙面提出を行った日時を出力
32	取下区分	withdrawalStatus	取下書は“1”，取り下げられた書類は“2”，それ以外は“0”を出力
33	書類情報修正区分	docInfoEditStatus	財務局職員が書類を修正した情報は“1”，修正された書類は“2”，それ以外は“0”を出力
34	開示不開示区分	disclosureStatus	財務局職員によって書類の不開示を開始した情報は“1”，不開示とされている書類は“2”，財務局職員によって書類の不開示を解除した情報は“3”，それ以外は“0”を出力
35	XBRL 有無フラグ	xbriFlag	書類に XBRL がある場合は“1”，それ以外は“0”を出力
36	PDF 有無フラグ	pdfFlag	書類に PDF がある場合は“1”，それ以外は“0”を出力
37	代替書面・添付文書有無フラグ	attachDocFlag	書類に代替書面・添付文書がある場合は“1”，それ以外は“0”を出力
38	英文ファイル有無フラグ	EnglishDocFlag	書類に英文ファイルがある場合は“1”，それ以外は“0”を出力

(出所：金融庁 [2019], 12-15 頁。一部省略)

①について、「date」はこちらが指定した日付であり、「type」は既に述べた通り提出書類一覧である②の情報を得る動作を指定している。「count」は、有価証券報告書だけでなく指定日に提出された書類の数を示しており、出力結果2の場合、2020年4月1日に255件の書類が提出されたことがわかる。

次にステータスについては出力内容を【表3】に示す。

表3 ステータスの内容

ステータス	メッセージ	説明	対処方法
200	OK	リクエスト成功時	-
400	Bad Request	リクエストデータに問題あり	リクエスト内容が誤っている。リクエスト内容（エンドポイント、パラメータの形式等）を見直す。
404	Not Found	リソースが存在しない	データが取得できない。パラメータの設定値を見直す。
500	Internal Server Error	サーバ側での処理中にエラー	EDINETのトップページ又は金融庁のウェブサイトの各種情報検索サービスにメンテナンス等の情報を確認する。

(出所：金融庁 [2019], 45頁。)

出力結果2のステータスをみると「200」となっているため、このリクエストは成功している。他のステータスが出力された場合、対処方法に従ってリクエストを修正する必要がある。

次に②の主な情報についてみていく。連番 (seqNumber) は検索結果の表示順をしめしている。出力結果2では255件の書類が検索されたが、連番の値が1であれば、1番目の書類、2であれば2番目の書類を指す。なお、連番は基本的に書類の提出日時順に付されるようになっている。

書類管理番号 (docID) は、XBRLデータを扱うのであれば重要な情報である。XBRLデータは、書類管理番号と同じ名前のフォルダに保存されている。どのフォルダがどの会社のいつの書類かを探すためには、書類管理番号をたどることが有効になる。

EDINET コードや証券コードは、API の登場以前では書類と対応させることが困難であった。先のように書類は書類管理番号で分類されているが、会社は EDINET コードや証券コードによって分類されることが多い。従来これらのコードと書類管理番号の紐づけが求められていたが、自作のプログラムに頼ることで紐づけを実現されていた。

府令コード、様式コード、書類種別コードは情報の整理に適している。例えば、有価証券報告書を検索したい場合、従来でも書類別で「有価証券報告書」を選択することは可能であった。しかし、EDINET に提出される膨大な書類をすべて検索してから、それぞれの書類に分類できることは、データの前処理の観点からも作業が容易となる。

次に時間データに関しても項目が設定されている。会計書類では期首と期末の情報は特に重要である。日本だけでなく、世界の企業は会計期間を設定しているが、この会計期間は基本的に企業の任意の開始日（期首）を決めることができる。日本では6割強の企業が4月1日を期首としており、その他の企業もおおむね月初め1日を期首としている。一方で、アメリカでは、期首として設定できない日を除き、1年365日に分散している。このような分析も以前であれば Web ページを解析して取得する作業であったが、API から提供されるデータとなった。提出日時についても同様で、中溝（2018）ではこのデータの取得を行ったが、API へのリクエストで取得できるようになったことは喜ばしいことである。

XBRL 有無フラグや PDF 有無フラグを使えば、XBRL データと PDF データをまとめて取得することもできる。同じフォルダの中に XBRL データと PDF データを保存することは、単純な作業では実現しなかったが、今回のフラグを利用することで、可能となる。しかし、この作業は、API の機能というよりは、プログラムで一工夫するような作業になる。

第4節 書類取得 API

本節では書類取得 API について説明する。書類一覧 API と同様に、リクエストとレスポンスがある。書類取得 API についてもエンドポイントが設定されている。

```
https://disclosure.edinet-fsa.go.jp/api/バージョン /
documents/書類管理番号
```

バージョンは「v+整数」で表記する。2020年4月時点ではバージョンは1であるため、「v1」を用いる。よって、本稿で使用するエンドポイントは次のようになる。

```
https://disclosure.edinet-fsa.go.jp/api/v1/documents/書類管理番号
```

書類管理番号とは、前節で取り上げた【表2】のNo.13の情報である。このエンドポイントに、【表4】のリクエストパラメータを組み合わせることでリクエスト URL が完成する。

表4 書類取得 API のリクエストパラメータ

パラメータ名	項目名	必須	設定値	説明
type	必要書類	○	1	提出本文書及び監査報告書を取得する。
			2	PDFを取得する。
			3	代替書面・添付文書を取得する。
			4	英文ファイルを取得する。

例えば、株式会社ニトリホールディングスの2019年5月17日に提出された第47期の有価証券報告書をリクエストしてみる。この有価証券報告書の書類管理番号が「S100FSYO」であり⁶⁾、リクエストパラメータの値を「2」とすると、有価証券報告書のPDFを閲覧する書類取得APIのリクエストURLは次の通りになる。このURLをWebブラウザに入力すると、PDF形式の有価証券報告書が画面に表示される。

```
https://disclosure.edinet-fsa.go.jp/api/v1/documents/  
S100FSYO?type=2
```

また、リクエストパラメータの値を「1」にすると、XBRLデータのダウンロードが始まる。XBRLデータはzip形式という圧縮データで取得するため、データの再利用をするには解凍をしなければならない。本稿では取り上げていないが、他の「3」や「4」の値を用いた場合もzip形式のファイルが提供されるようになっている。

```
https://disclosure.edinet-fsa.go.jp/api/v1/documents/  
S100FSYO?type=1
```

なお、リクエストパラメータの値を「1」と「2」を並列させ、「type=1&type=2」のようにしても、XBRLデータとPDFデータを同時に取得することはできず、エラーが出力される。

6) 書類管理番号は事前に書類一覧APIで調べていることを前提としている。第3節のコード2で示したリクエストURLの日付を「2019-05-17」とすると、連番228に情報が記載されている。

第5節 お わ り に

本稿では、EDINET APIについて、その内容について仕様書を中心にみてきた。EDINET APIは従来のように自作プログラムでアクセスするよりも、効率が上がっている。特に、規則性のあるリクエスト URL を与えられたことで、レスポンスの中から必要な情報を取捨選択し、情報が整理された上で保存できることが大きい。データ分析では、前処理の段階で全作業の大半の時間を要すると言われている。XBRL データもデータ数を揃えることが困難であったため、API を利用することで XBRL データの取得難易度が下げられたと言える。

一方で、本稿で示したリクエスト URL の内容だけでは、膨大なデータを取得することはできない。例えば Python のようなプログラミング言語と組み合わせることで、それが可能となる。会計領域の人にとって、プログラミングが浸透しているとは言えず、API の利用には時間を要すると予想される。しかし、EDINET API は非常に使いやすく、この金融庁のような国の機関が公式で API を提供している環境は世界でも稀である。Python を一から学び、本稿の内容を利用するところまでは、(容易とまでは言わないが) ハードルがとても高いわけではない。XBRL データを利用した分析が増えることを期待する。

参 考 文 献

- 加藤耕太 [2017], 『Python クローリング&スクレイピングーデータ収集・解析のための実践 開発ガイドー』技術評論社。
- 金融庁 [2019], 「EDINET API 仕様書 (Versiton 1)」 (<https://disclosure.edinet-fsa.go.jp/EKW0EZ0015.html>)。
- 中溝晃介 [2018], 「クローリング EDINET」松山大学論集, 第 30 巻第 4-2 号, 143-162 頁。

本稿は、2018 年度に交付を受けた松山大学特別研究助成による研究成果の一部である。