

大学組織データ分析システム開発のための Web API Web API for development of analysis system for university institutional data

井田 正明[†]
Masaaki Ida

1. はじめに

高等教育の質保証では、組織に関する多量の教育・研究・財務等のデータの分析および分析システムの開発について、その知識と実務開発経験が重要である。そのため、基礎となる大学等の組織データの効果的な収集・整備が課題となる。またそのデータの質管理も重要である。このような課題への対処として、本稿では、これまで研究開発してきた大学組織データに関する Web API について述べる。

ここでのユーザは、社会科学・教育学系のデータ分析者を想定としている。データ分析システムを開発するにはデータの整備と管理が必要となり、これには多大な時間と労力を要する。これら課題が解決できれば、データベースの設計・管理の専門家ではない分析者が、大学改革や財務等の特定の専門分野についてのデータ分析とその解釈過程に十分に注力することができるようになる。このような課題への対処方法のひとつとして、データベースの Web API 化がある。Web API はネットワークを介して情報処理を行うための機能とその形式の取り決めである。分析ユーザから分離された分散型データベースとみなすことができる。Web API によって、最新に整備されたデータベースから、必要となしに必要部分のデータを取得することが可能となる。またユーザが個別に有するデータベースや他の Web API と組み合わせることによって、より価値のあるデータ分析を行うことが可能となる。

2. 大学基本情報とその Web API

日本の高等教育機関の学生数・教職員数などのデータは、総務省統計局の e-Stat Web API [1] から長期データを得ることができる。ただし e-Stat が提供するデータは集計データであり、個別の大学・学部の詳細なデータではない。

大学改革支援・学位授与機構の Web サイトにおいては、国公立大学・短期大学の大学基本情報を Excel 形式のファイルによって公開している[2]。データの内容は、表 1 のような大きく 7 つの分類 (7 号票, 8 号票, 9 号票等) であり、さらに表の右側のように内部が小区分に分かれている。

表 1 大学基本情報の一覧[2]

学生教職員等 (7)	(7-A) 学生数
	(7-B) 教員数 (本務者)
	(7-1) 教員数 (本務者) (再掲)
	(7-Z) 教員数 (兼務者)
	(7-C) 職員数
	(8-D) 学科別学生数 入学志願者数 入学者数

[†] 大学改革支援・学位授与機構, National Institution for Academic Degrees and Quality Enhancement of Higher Education (NIAD-QE)

学部学生内訳 (8)	(8-2) 学科別学生数のうち休学者数
	(8-3) 学科別学生数のうち最低在学年限超過学生数 (編入学者は除く)
	(8-G) 出身高校の所在地県別入学者数
	(8-O) 年齢別入学者数 (再掲)
	(8-7) 専攻科, 別科及び科目等履修生等の学生
	(8-R) 短期大学・高等専門学校・専修学校 (専
大学院学生内訳 (9)	(9-H) 専攻別学生数 左記のうち社会人
	(9-4) 専攻別学生数のうち休学者数
	(9-5) 専攻別学生数のうち最低在学年限超過学生数 (編入学者は除く)
	(9-I) 入学状況 入学志願者数 入学者数
	(9-S) 年齢別入学者数
	(9-8) 科目等履修生等の学生数
本科学生内訳 (10)	(10-J) 学科別学生数 入学状況 (本科)
	(10-6) 学科別学生数のうち休学者数
	(10-K) 出身高校の所在地県別入学者数
	(10-Q) 年齢別入学者数 (再掲)
	(10-9) 専攻科, 別科及び科目等履修生等の学
	(10-T) 高等学校等専攻科からの編入学者数
外国人学生 (11)	(11) 国費留学生, 私費留学生, 留学生以外の
	(11 別掲) 国費留学生, 私費留学生, 留学生以外の外国人学生 (専攻科・別科の学生, 科目等履修生・聴講生・研究生)
学校施設 (20)	(20) 学校施設
卒業後の状況 (30)	(30-1) 状況別卒業生数, 入学年度別卒業生数
	(30-2) 職業別就職者数, 産業別就職者数

これらのデータを分析するためには、多数存在するデータファイルを整備するための相当な時間と労力を必要とする。これまでにこれらデータの Web API 化は、当機構研究開発部において検討を進めてきた。その後、2022 年より、大学ポータルより Web-API 機能の提供が始められた (<https://portraits.niad.ac.jp/>)。しかしながら、現状では数年度分のデータ提供であり、また、号票・年度ごとの単位となっている。

これまでに、表 1 の一部については Web API 化を行ってきた。本稿では、表 1 の 7 つの分類全体について、データ

分析のために容易に利用が可能な Web API 開発の現状について述べる。

ここでの Web API は、REST 形式であり JSON 形式で結果を戻すものである。それぞれの Web API の単位は、表 1 の右側にある (7-A), (7-B) といった小区分である。そのためエンドポイントとなる各 URL は、G に各表記号を付与して対応させている。たとえば、(8-D)についてはつぎである。

```
https://( Web API server URL )/api/G08D
```

利用方法として、例えば、Python のプログラムでの Web API の利用は、つぎを想定している。

```
response = requests.get(url, params=params, headers=headers)
if response.status_code == 200:
    data = response.json()
```

API のパラメータは (apikey に加えて) 共通としてつぎである。

- ・ limit : 最大の出力データ数
- ・ from : データの開始年度 (西暦)
- ・ to : データの終了年度 (西暦)
- ・ gkcode : 学校コード
- ・ unname : 大学名 (gkcode と同内容)

これら共通のパラメータに加え、各 Web API で分析に必要なパラメータを追加している (たとえば、学部・研究科コード、学科・専攻科コード、分野別、昼夜別等)。つぎはパラメータの例である。

```
url = "https://*****/api/G08D"
params = {
    'apikey': '*****',
    'limit': '10',
    'unname': '* * 大学',
    'fcode': '1G01', # 工学部
    'dcode': '9999', # 学科の合計
    'from': '2019', # 開始年度
    'to': '2023',
}
```

出力例はつぎとなる。構造は大学基本情報の小分類のデータの全てについて、'data' 以下に階層を一段とした構造としている。

```
{'data': [
    {'年度': '2022',
     '学校コード': 'F*****',
     '学部番号': '1G01',
     '所在地': '53',
     '大学名': '****大学',
     '昼夜別': '1',
     '学部名': '工学部',
     '学科': '計',
     '符号': '9999',
     '1年_男': '881',
     '1年_女': '97',
     '2年_男': '879',
     '2年_女': '100',
     (途中省略)
     '計_男': '3767',
```

```
'計_女': '407',
'志願者_男': '2355',
'志願者_女': '330',
'入学者_男': '881',
'入学者_女': '97'},
{'年度': '2023',
 (以下, 同様)
 *****,
```

3. Web API の利用と開発における課題

本稿での想定は、大規模なデータ分析システムの開発ではなく (その場合は、事前に全データを取得し内部 DB を構築するのが望ましい)、社会科学・教育学系のユーザが、Python (や BI システム, Excel 等) を使って、容易に開発が可能な分析システムを想定している。以下では、Web API の利用と開発における課題を述べる。

まず、Web API のデータ・指標の種類とそれぞれの構造の理解のしやすさは重要となる。ここでは、表 1 の大学基本情報の小区分を Web API の単位としており、個々の Web API のデータ構造は、エクセル形式をほぼそのまま利用したものとなっている。また、このような構造の理解のしやすさは、長期にわたる Web API の管理も容易となる。

つぎに、パラメータの項目設定については分析の内容に依存するが、利用想定としてはつぎである。

- ・ ある年度において、様々な単位の複数の組織を複数の指標で分析する。
- ・ ある組織について、複数の指標の経年変化を分析する。そのため、階層別でのグループ化 (例えば、規模別) が可能となるような、わかりやすいコード整備が必要となる。また、Web API のレスポンス速度は、データの通信量に依存するため、無駄を省き分析に必要な部分のみを適切に抽出できるパラメータ構成で、できるだけ少数回のリクエストで、通信コストを低減できるものであることが望ましい。
- また、Web API からのレスポンス (JSON) を Python (や BI システム, Excel 等) の分析環境に容易に取り込むことができ (データフレーム等への変換を含めて)、さらに他のデータソースとの連携も容易であることが望ましい。
- さらに、大学基本情報の固有の課題として、以前の古いデータと最近のデータとの項目や構造の差にある程度対応できる必要がある。

4. おわりに

本稿においては、社会科学等における簡便なデータ分析システムを開発するためのデータ取得・管理方法として、大学基本情報の Web API の研究開発を述べた。データ分析自体については他の機会に述べる予定である。現在、他の Web API (財務情報等) との連携による、柔軟なデータ分析および新指標の開発についての研究開発を進めている。

参考文献

- [1] e-Stat API 機能 : <https://www.e-stat.go.jp/api/>
- [2] 大学基本情報, <https://portal.niad.ac.jp/prtr/table.html> (大学改革支援・学位授与機構「大学基本情報」をもとに本稿の Web API を作成)