

九工大アプリ2018年度版サーバーリファレンス

Introduction

このドキュメントは九工大アプリ2018年度版のAPI仕様書です。
現段階では情報工学部限定、かつ学部生の使用をメインに据えて考えています。

文書の構成として以下のようになっています。

- 1. Explanation of Model
- 2. Explanation of URL
- 3. Enviroments
- 4. How to Execute
- 5. Thanks

できるだけの確認はしましたがエラーの可能性あり。

速やかな連絡を希望します。

1. Explanation of Model

ここではこのサーバーで利用する主なJSONモデルについて説明します。

このサーバーは以下の5つのモデルから成り立っています。

- NewsHeading
- News
- Syllabus
- User
- UserSchedule

1.1 NewsHeading

お知らせ情報の見出しを表すモデル

column

Name	Type	Description	Example
news_heading_code	int	NewsHeadingを識別するためのID.	357
short_name	string	一覧表示するときの一文字	知
name	string	名前	お知らせ(学生向け)
color_code	string	色コード	#219653
updated_at	string	お知らせが最後に更新された時の時刻	2018/04/15 19:15

1.2 News

お知らせを表すモデル

column

Name	Type	Description	Example
infos	object	各種情報が入れ子となっている	"infos": {,,,}
attachement_infos	object	添付資料の情報が入れ異なっている	"attachement_infos: {,,,}

infosの説明

infosの中では各情報のkey=タイトルとvalue=内容の辞書となって格納されています。詳細はURLにて。

attachement_infosの説明

attachement_infosの中では各添付資料の資料番号(添付資料2など)と内容が対となって格納されています。
以下の2角要素がひとつのattachement_infoを形成しています。

Name	Type	Description	Example
title	string	添付資料のタイトル	講義室変更について.pdf
url	string	添付資料のダウンロード先URL	https://aiueo/a.pdf

1.3 Syllabus

授業情報を表すモデル

column

Name	Type	Description	Example
title	string	授業名	解析I
subject_code	int	授業に割り当てられているコード番号。ほぼ使うことはないでしょう。	11003000
teacher_name	string	この授業の講師の名前。複数人いる場合もある。	山田花子
target_participants_infos	object	授業対象者の主に単位に関する情報を一覧表示する。	target_participants_infos: {,,,}
target_school_year	string	対象学年	1, 3年
class_number	int	それぞれの授業に割り振られている番号。同じ授業を複数人で扱っている際に使用される。	1
target_period	string	授業の開講される日にち、時間帯	火曜3限,木曜3限
published_date	date	シラバス情報の更新された日にち	2018/02/22 00:00
abstract	string	授業概要	長いので省略
positioning	string	授業の位置づけ	長いので省略
lecture_content	string	授業の進行予定	長いので省略
lecture_processing	string	授業の進行方針	長いので省略
performance_target	string	この授業を学んで何が得られるか	長いので省略
valuation_basis	string	点数評価の対象	長いので省略
instruction_out_learning	string	授業外ですべきこと、書いてないことが多い。	長いので省略
keywords	string	授業に関連するキーワード	長いので省略
text_books	string	授業で使う教科書の名前	長いので省略
study_aid_books	string	授業で必要になるかもしれない参考書の名前	長いので省略
notes	string	備考, ほぼ書かれてない	長いので省略
professor_email	string	担当教授のメアド	aiueo@aiueo.com

target_participants_infosの説明

このobjectのリスト内では対象学生に対して割り当てられている要習得単位数や単位の種類が書かれています。

以下の3つの要素がひとつのobjectを形成しています。

Name	Type	Description	Example
target_participants	string	対象学生の所属学科名	情報工学部 知能情報工学科 知能情報工学科 (編入)
academic_credit_kind	string	対象学科においてこの授業がどのような扱いの単位なのかを示す	査定外
academic_credit_num	float	対象学科においてこの授業を履修することでいくつ単位をもらえるのかを示す	0.0

1.4 User

利用ユーザーを表す

column

Name	Type	Description	Example
pk	int	primary Key	1
school_year	int	学年を表す。0: 1年, 1: 2年, 2: 3年, 3: 4年というふうに一つ数字がずれているので注意	2
department	int	どの学科に属するかを示す。数字がどこに対応するかは下表にて記す。	23

departmentの対応表

departmentの値	対応学科
21	情報工学部 知能情報工学科
22	情報工学部 情報・通信工学科
23	情報工学部 知的システム工学科
24	情報工学部 物理情報工学科
25	情報工学部 生命情報工学科

1.5 UserSchedule

各ユーザーの一週間の履修予定授業を表すモデル

1データがあるユーザーの履修予定授業1コマを表します。

column

Name	Type	Description	Example
syllabus	Syllabus	履修教科を示す	Syllabusの項を参照
day	int	履修曜日を表す 0: Mon, 1: Tues, ,, , 4: Fri と行った感じ	3
period	int	履修時限を表す 0: 1限, 1: 2限, ,, , 4: 5限 と行った感じ	2
quarter	int	履修した科目は何クォーターで履修しているかを表す。0: 1クォーター, 1: 2クォーター, 2: 3クォーター, 3: 4クォーターとなる	3

2. Explanation of URL

ここではAPIにおいて使用できるエンドポイントを説明します。
いちおうRESTに基づいて作成してますが仕様上必要のないエンドポイントは省略しています。

BASE_URL : localhost:8000/api

2.1 List Of API

EndPoint	HTTP Method	description	request data_type	response_type
/users/	POST	ユーザーの作成	json	json
/users/	PUT	ユーザーの更新	json	json
/users/<int:user_id>/	GET	ユーザー詳細取得	json	json
/news-headings/	GET	Newsの見出し一覧	None	json
/news/code-<int: code>	GET	あるNewsHeadingに紐付いたNewsの一覧取得	None	json
/syllabuses/	GET	授業情報一覧、膨大なデータ量なので基本使っちゃダメ	None	json
syllabuses/<int:syllabus_id>	GET	シラバス詳細取得	None	json
syllabuses/day-<int: day>/period-<int:period>	GET	ある日にち、時限におけるシラバス一覧取得	None	json
/user-schedules/	POST	ユーザーの時間割の作成	json	json
/user-schedules/user-<int: user_id>/quarter-<int: quarter_id>	GET	あるユーザーが設定したnクォーターにおける時間割の取得	None	json

2.2 Details Of API

2.2.1 POST /users/

ユーザーを作成します。初めてアプリを開いた時のみ呼ばれるはず。

```
HTTP/1.1 201

Content-Type : application/json

Request json:
{
  "school_year": 3,
  "depart_ment": 2
}

Response json:
{
  "pk": 8,
  "school_year": 1,
  "department": 25,
  "created_at": "2018/05/02 15:18",
  "updated_at": "2018/05/02 15:18"
}
```

2.2.2 PUT /users/

ユーザー情報を更新します。学年が変更したとか学科転向したとか信仰上の問題とか。

```
HTTP/1.1 200

Content-Type : application/json

Request json:
{
```

```
    {
      "pk": 5,
      "school_year": 1,
      "department": 22,
      "created_at": "2018/05/01 21:41",
      "updated_at": "2018/05/01 21:41"
    }

Response json:
{
  "pk": 5,
  "school_year": 1,
  "department": 22,
  "created_at": "2018/05/01 21:41",
  "updated_at": "2018/05/02 16:24"
}
```

2.2.3 GET /users/<int:user_id>

ユーザーID = user_idのユーザーの情報を取得します。

```
Example url : /api/users/5

HTTP/1.1 200

Content-Type : application/json

Response json:
{
  "pk": 5,
  "school_year": 1,
  "department": 22,
  "created_at": "2018/05/01 21:41",
  "updated_at": "2018/05/02 16:24"
}
```

2.2.4 GET news-headings/

Newsの見出し情報を一覧取得します。

```
HTTP/1.1 200

Content-Type : application/json

Response json:
[
  {
    "news_heading_code": 357,
    "short_name": "知",
    "name": "お知らせ(学生向け)",
    "color_code": "#219653",
    "updated_at": "2018/04/15 19:15"
  },
  ,,, (以下省略)
]
```

2.2.5 GET news/code-<int: news_heading_code>/

あるNewsHeadingに紐付いたNews情報を一覧取得します。
codeとしてはNewsHeading一覧から取得した news_heading_code を使用します。

```
Example url : /api/news/code-354/

HTTP/1.1 200

Content-Type : application/json

Response json:
[
  {
    "news_heading": 86,
    "infos": {
      "タイトル": "講義室変更",

```

```
        "科目名": "授業A",
        "教員名": "山田先生",
        "対象学科": "大学院",
        "対象学年": "大学院",
        "日付": "2018年5月31日",
        "時限": "2限",
        "種別": "変更日時以降",
        "変更前": "E106講義室",
        "変更後": "1101講義室",
        "内容": ""
    },
    "attachement_infos": {
        "添付資料(1)": {
            "title": "講義室変更.pdf",
            "url": "https://aiueo/a.pdf"
        }
    }
},
,,, (省略)
]
```

2.2.6 GET /syllabuses/

情報工学部のシラバスの情報を一覧取得します。授業情報をすべて取得するため、動きが重い。用意したが使わないのが無難。

```
HTTP/1.1 200

Content-Type : application/json

Response json:
[
  {
    "title": "授業A",
    "subject_code": 11000300,
    "teacher_name": "山本 花子",
    "target_participants_infos": [
      {
        "target_participants": "情報工学部 情工1類 Iクラス",
        "academic_credit_kind": "査定外",
        "academic_credit_num": 2.0
      },
      {
        "target_participants": "情報工学部 情工1類 IIクラス",
        "academic_credit_kind": "査定外",
        "academic_credit_num": 2.0
      },
      {
        "target_participants": "情報工学部 情工2類 IIIクラス",
        "academic_credit_kind": "査定外",
        "academic_credit_num": 2.0
      },
      {
        "target_participants": "情報工学部 情工3類 IVクラス",
        "academic_credit_kind": "査定外",
        "academic_credit_num": 2.0
      },
      {
        "target_participants": "情報工学部 情工3類 Vクラス",
        "academic_credit_kind": "必",
        "academic_credit_num": 2.0
      }
    ],
    "target_school_year": "1年",
    "target_term": "前期",
    "class_number": 1,
    "target_period": "集中講義",
    "published_date": "2018/03/07 00:00",
    "abstract": "授業概要 たいてい長い",
    "positioning": "",
    "lecture_content": "授業の今後の方針 \nで区切っているのでよしなにどうぞ",
    "lecture_processing": "教科書を基にした講義形式で進める。",
    "performance_target": "",
    "valuation_basis": "期末テスト 100%",
    "instruction_out_learning": "授業以外でも教科書に1日最低5時間目を通すこと",
    "keywords": "python, haskell, kotlin, scala, matz",
    "text_books": "みんなのH本",
    "study_aid_books": "学習の助けになる本",
    "notes": "備考 たいてい書いてない",
    "professor_email": "aiueo@ab.jp"
  },
  ...
]
```

```
    ..., (省略)
  ]
}
```

2.2.7 GET /syllabuses/<int: syllabus_id>

primary key = syllabus_id であるsyllabusを取得する。
余り使わないと思う。

```
URL Example: /syllabuses/2018/

HTTP/1.1 200

Content-Type: application/json

Response json:
{
  "title": "授業A",
  "subject_code": 11000300,
  "teacher_name": "山本 花子",
  "target_participants_infos": [
    {
      "target_participants": "情報工学部 情工1類 Iクラス",
      "academic_credit_kind": "査定外",
      "academic_credit_num": 2.0
    },
    {
      "target_participants": "情報工学部 情工1類 IIクラス",
      "academic_credit_kind": "査定外",
      "academic_credit_num": 2.0
    },
    {
      "target_participants": "情報工学部 情工2類 IIIクラス",
      "academic_credit_kind": "査定外",
      "academic_credit_num": 2.0
    },
    {
      "target_participants": "情報工学部 情工3類 IVクラス",
      "academic_credit_kind": "査定外",
      "academic_credit_num": 2.0
    },
    {
      "target_participants": "情報工学部 情工3類 Vクラス",
      "academic_credit_kind": "必",
      "academic_credit_num": 2.0
    }
  ],
  "target_school_year": "1年",
  "target_term": "前期",
  "class_number": 1,
  "target_period": "集中講義",
  "published_date": "2018/03/07 00:00",
  "abstract": "授業概要 たいてい長い",
  "positioning": "",
  "lecture_content": "授業の今後の方針 \nで区切っているのでよしなにどうぞ",
  "lecture_processing": "教科書を基にした講義形式で進める。",
  "performance_target": "",
  "valuation_basis": "期末テスト 100%",
  "instruction_out_learning": "授業以外でも教科書に1日最低5時間目を通すこと",
  "keywords": "python, haskell, kotlin, scala, matz",
  "text_books": "みんなのH本",
  "study_aid_books": "学習の助けになる本",
  "notes": "備考 たいてい書いてない",
  "professor_email": "aiueo@ab.jp"
}
```

2.2.8 GET /syllabuses/day-<str: day>/period-<int: period>/

ある曜日、ある時限において開講されている授業を一覧取得します。

dayの対応表を以下に示す。

day_code	曜日
monday	月曜
tuesday	火曜

wednesday	水曜
thursday	木曜
friday	金曜

一意に決まるならば `m` のように省略しても良い。

`period`については $(\text{period} + 1)$ 限目を取得する。

つまり、火曜3限の授業一覧を取得したければ、

`/syllabuses/day-tue/period-2/`

を叩けば良い。

取得するデータの構成は **2.2.6 GET `/syllabuses/`** とおなじため省略。

2.2.9 POST `/user-schedules/`

ユーザーが設定した時間割のある1コマの情報を登録する。

`day`は0: 月曜, 1: 火曜, 2: 水曜, 3: 木曜, 4: 金曜に対応しています。

また、`quarter_id`は 0 ~ 3までの数字であり、 $(\text{quarter_id} + 1)$ クォーターとして表現されます。

下例は月曜4限第1クォーターに授業を登録したい場合です。

```
HTTP/1.1 201

Content-Type : application/json

Request json:
{
  "user_id": 344,
  "syllabus_id": 2018,
  "day": 0,
  "period": 3,
  "quarter": 0
}

Response json:
{
  "user": {
    /users/ と同等
  },
  "syllabus": {
    /syllabuses/ と同等
  },
  "day": 0,
  "period": 3,
  "quarter": 0
}
```

2.2.10 GET `/user-schedules/user-<int: user_id>/quarter-<int: quarter_id>`

あるユーザーの n クォーターにおける時間割を返します。

UserScheduleオブジェクトのリストとして表現しています。

`quarter_id`は 0 ~ 3までの数字であり、 $(\text{quarter_id} + 1)$ クォーターとして表現されます。

```
HTTP/1.1 200

Content-Type : application/json

Response json:
[
  {
    "user": {
      /users/ と同等
    },
    "syllabus": {
      /syllabuses/ と同等
    },
    "day": 0,
    "period": 3,
    "quarter": 0
  },
  ...
]
```



```
        "syllabus": {
            /syllabuses/ と同等
        },
        "day": 0,
        "period": 3,
        "quarter": 0
    },
    ..., (省略)
]
```

3. Enviroments

- Python 3.6.3
- module類

beautifulsoup4==4.6.0
Django==2.0.4
django-extensions==2.0.6
django-filter==1.1.0
djangorestframework==3.8.0
pytz==2018.3
selenium==3.11.0
six==1.11.0

4. How to Execute

python 3.6.3, pip 9.0.3が入っている前提ですすめる。

初期化処理についてのみ説明する。

1. projectのrootに移動。
2. `pip install -r requirements.txt`
を実行し、必要なモジュール群をinstallする。
3. `python manage.py initialize_news`
を実行し、学生のお知らせ情報を取得する。
4. `python manage.py initialize_syllabus`
を実行し、授業情報を取得する。

`python manage.py scrape_news` とたたくことでNewsを最新の状態に更新できる。