
classtime documentation

Release 0.1

Ross Anderson and Andrew Hoskins

March 23, 2016

1	The API	3
1.1	Versioning	3
1.2	Pagination	3
1.3	Search queries	4
1.4	Formats used in responses	4
1.5	api/v1/institutions	5
1.6	api/v1/terms	6
1.7	api/v1/courses-min	6
1.8	api/v1/courses/<course>	7
1.9	api/v1/generate-schedules	9
1.10	api/v1/schedules	12

Classtime is an HTTP API for course data and schedule generation at UAlberta.

Purpose: “Build a university schedule that fits your life in less than five minutes”

It can be used for the following:

- browse terms
- browse courses
- get details on any course
- generate schedules, with support for core courses, electives, and preferences

Classtime currently only supports the University of Alberta.

Documentation:

The API

Responses are communicated in [JavaScript Object Notation \(javascript\)](#). Each endpoint returns a list of objects. A few useful book-keeping items are also included in each response.

```
{
  "num_results": <int>,
  "objects": [
    {
      <key>: <value>,
      <key>: <value>,
      ...
      <key>: <value>
    },
    { <response object 2> },
    ...
    { <response object N> }
  ],
  "page": <int>,
  "total_pages": <int>
}
```

The exception is `api/v1/courses/<course>`, which returns a single object (not a list), and no book-keeping items.

It is possible for zero `<response object>`s to be returned.

1.1 Versioning

Versions should be assumed incompatible with one another. Versions are prefixed with their version number. A maximum of 2 versions will be supported at any one time.

Current versions, with their prefixes.

- `v0 /api/<endpoint>` (deprecated)
- `v1 /api/v1/<endpoint>`

1.2 Pagination

Each response includes:

- `page` := page number returned
- `total_pages` := total number of pages

To get the *n*th page, append `?page=<n>` to any endpoint:

```
GET /api/v1/courses-min?page=2
```

If you are using a search query, append the page number with `&`:

```
GET /api/v1/courses-min?q=<search_query>&page=2
```

```
Compression  
~~~~~
```

All responses are compressed with gzip. Your client should handle the gzipping automatically for you

1.3 Search queries

Search queries are used to restrict an endpoint's output. This is useful both for performance and semantic reasons.

Format is:

```
/api/v1/<endpoint>?q={"filters":[{"name":<attribute_name>,"op":<comparison>,"val":<attribute_value>},
```

Examples:

- Get courses for a certain institution and a certain term:

```
GET /api/v1/courses-min?q={"filters":[{"name":"institution","op":"equals","val":"ualberta"},{"na
```

- Get terms for a certain institution:

```
GET /api/v1/terms?q={"filters":[{"name":"institution","op":"equals","val":"ualberta"}]}
```

Available operators [listed here](#). As of this writing, they are:

```
==, eq, equals, equals_to  
!=, neq, does_not_equal, not_equal_to  
>, gt, <, lt  
>=, ge, gte, geq, <=, le, lte, leq  
in, not_in  
is_null, is_not_null  
like  
has  
any
```

1.4 Formats used in responses

1.4.1 Day format

String containing one or more of the characters “MTWRF”, with each corresponding to a day from Monday through Friday.

eg "MWF"

eg "TR"

1.4.2 Time format

"HH:MM XM"

HH 2-digit hour between 00 and 12

MM 2-digit minute between 00 and 59

X A or P

eg "08:00 AM"

eg "09:50 PM"

1.5 api/v1/institutions

Retrieve a list of available institutions. Each institution contains all available information.

1.5.1 Request

```
GET localhost:5000/api/v1/institutions
```

1.5.2 Response

```
{
  "objects": [
    {
      "institution": "ualberta",
      "name": "University of Alberta"
    },
    { <institution object 2> },
    ...
    { <institution object N> }
  ]
  ...
}
```

objects list of <institution object>s

<institution object>

institution variable length institution identifier

name semantic institution name

1.6 api/v1/terms

Retrieve a list of available terms. Each term contains all available information.

1.6.1 Request

```
GET localhost:5000/api/v1/terms
```

1.6.2 Response

```
{
  "objects": [
    {
      "endDate": "2007-12-05",
      "startDate": "2007-09-05",
      "term": "1210",
      "termTitle": "Fall Term 2007"
    },
    { <term object 2> },
    ...
    { <term object N> }
  ],
  ...
}
```

objects list of <term object>s

<term object>

endDate YYYY-MM-DD

startDate YYYY-MM-DD

term 4-digit term identifier

termTitle semantic term name

1.7 api/v1/courses-min

Quickly retrieve a hierarchy of available courses.

Each course object contains only essential information. More detailed information about a specific course is retrieved with */api/v1/courses*.

1.7.1 Request

```
GET localhost:5000/api/v1/courses-min
```

1.7.2 Response

```
objects = [
  {
    "faculty": "Faculty of Business",
    "subjects": [
      {
        "subject": "ACCTG",
        "subjectTitle": "Accounting",
        "courses": [
          {
            "course": "000001",
            "asString": "ACCTG 300",
            "courseTitle": "Intermediate Accounting"
          },
          { <course object> }
          ...
        ]
      },
      { <subject object> }
      ...
    ]
  },
  { <faculty object> }
  ...
]
```

objects list of *faculty objects*

<faculty object>

faculty semantic faculty name

subjects list of *subject objects*

<subject object>

subject variable-length subject identifier

subjectTitle semantic subject name

courses list of *course-min objects*

<course-min object>

course 6-digit course identifier

asString <subject> <level>

courseTitle semantic course name

1.8 api/v1/courses/<course>

Retrieve detailed information about a single course.

1.8.1 Request

```
GET localhost:5000/api/v1/courses/<course>
```

course *6-digit unique course identifier*

1.8.2 Response

```
{
  "asString": "ACCTG 300",
  "career": "UGRD",
  "catalog": 300,
  "course": "000001",
  "courseDescription": "Provides a basic understanding of accounting: how accounting numbers
    are generated, the meaning of accounting reports, and how to use accounting reports to
    make decisions. Note: Not open to students registered in the Faculty of Business. Not
    for credit in the Bachelor of Commerce Program.",
  "courseTitle": "Introduction to Accounting",
  "department": "Department of Accounting, Operations and Information Systems",
  "departmentCode": "AOIS",
  "faculty": "Faculty of Business",
  "facultyCode": "BC",
  "subject": "ACCTG",
  "subjectTitle": "Accounting",
  "term": "1490",
  "units": 3
}
```

asString <subject> <level>

career variable-length abbreviation of university program type (undergrad, grad, ..)

catalog catalog id

course *6-digit unique course identifier*

courseDescription often long description of the course

courseTitle semantic course name

department semantic department name

departmentCode variable-length department identifier

faculty semantic faculty name

facultyCode variable-length faculty identifier

subject variable-length subject identifier

subjectTitle semantic subject name

term *4-digit unique term identifier*

units integer weight of the course

1.9 api/v1/generate-schedules

1.9.1 Request

```
GET localhost:5000/api/v1/generate-schedules?q=<q>
```

```
q = {
  "institution": institution,
  "term": term,
  "courses": [course, course2, ..., courseN],
  "busy-times": [
    {
      "day": "[MTWRF]{1,5}"
      "startTime": "##:## [AP]M",
      "endTime": "##:## [AP]M"
    },
    { <busytime object_2> },
    ...
    { <busytime object_n> }
  ],
  "electives": [
    {
      "courses": [course, course2, ..., courseN]
    },
    { <electives object_2> },
    ...
    { <electives object_n> }
  ],
  "preferences": {
    "start-early": <integer>,
    "no-marathons": <integer>,
    "day-classes": <integer>,

    "current-status": <boolean>,
    "obey-status": <boolean>
  }
}
```

See the method `TestAPI.test_generate_schedules` in `tests/angular_flask/test_api.py` for concrete examples.

institution *unique institution identifier*

term *4-digit unique term identifier*

courses list of *6-digit unique course identifier*

busy-times (optional) list of `<busytime>` objects

electives (optional) list of `<electives>` objects

preferences (optional) specify the weight of each *preference*. There are sensible defaults.

<busytime object>

day day(s) which are busy. Uses *day format*

startTime time the user starts being busy. Uses *time format*

endTime time the user is not busy anymore. Uses *time format*.

<electives object>

courses list of *course identifiers*

One course from each <electives object>'s *courses* list will be present in each schedule.

Preferences

In *preferences*, each key's value is the preference's **weighting**. Positive, negative, and zero-valued weightings are described for each preference type.

There are sensible defaults for each preference, and all preferences are optional.

Currently supported preferences:

- **no-marathons**

- weight > 0 = avoid long stretches of classes in a row
- weight < 0 = prefer long stretches of classes in a row
- weight = 0 = no preference

- **day-classes**

- weight > 0 = prefer daytime classes
- weight < 0 = prefer night classes (5pm and on)
- weight = 0 = no preference

- **start-early**

- weight > 0 = prefer early starts
- weight < 0 = prefer late starts
- weight = 0 = no preference

> Note: *start-early* can be used in tandem with *busy_times* to specify *how* early to start

There is also:

- **current-status**

- a boolean: `true` or `false`
- specifies whether the open/closed and active/cancelled status of sections should be updated

- **obey-status**

- a boolean: `true` or `false`
- specifies whether the open/closed and active/cancelled status of sections should be respected when scheduling
- if true, closed or cancelled sections will not be scheduled

1.9.2 Response

```
{
  "objects": [
    {
      "sections": [
        {
          ...
          <course attributes>
          ...
          "class_": "62293",
          "component": "LEC",
          "day": "MWF",
          "startTime": "10:00 AM",
          "endTime": "10:50 AM",
          ...
          "section": "A02",
          "campus": "MAIN",
          "capacity": 0,
          "instructorUid": "jdavis",
          "location": "CCIS L2 190"
        },
        { <section object 2> },
        ...
        { <section object N> }
      ],
      "more_like_this": [<schedule-identifier>, <schedule-identifier>, ..]
    },
    { <schedule object 2> },
    ...
    { <schedule object M> }
  ],
  ...
}
```

objects list of *schedule objects*

<schedule object>

sections list of *section objects*

more_like_this list of *schedule identifiers*

<section object>

<course attributes> all attributes from the parent *course* object

class_ 5-digit unique section identifier

component section type identifier, often 'LEC', 'LAB', 'SEM', 'LBL'

day day(s) the section is on. Uses *day format*

startTime time the section begins. Uses *time format*

endTime time the section ends. Uses *time format*

section section identifier. usually a letter and a number

campus variable-length campus identifier

capacity number of seats

instructorUid instructor identifier

location semantic location name

<schedule-identifier>

schedule-identifier variable length unique schedule identifier. Details about the schedule can be obtained by accessing *api/v1/schedules* and passing in this identifier.

1.10 api/v1/schedules

1.10.1 Request

```
GET localhost:5000/api/v1/schedules/<schedule-identifier>
```

course *schedule identifier*

1.10.2 Response

```
{
  "hash_id": "48c3df652685a23acd9a759b91f25b",
  "institution": "ualberta",
  "term": "1490",
  "sections": [
    {
      "asString": "ENGG 100 LEC A2",
      "autoEnroll": null,
      "campus": "MAIN",
      "capacity": 516,
      "classNotes": null,
      "classStatus": "A",
      "classType": "E",
      "class_": "61383",
      "component": "LEC",
      "course": "004093",
      "day": "R",
      "endTime": "01:50 PM",
      "enrollStatus": "O",
      "institution": "ualberta",
      "instructorUid": null,
      "location": "CCIS 1 430",
      "schedule": null,
      "section": "A2",
      "session": "Regular Academic Session",
      "startTime": "01:00 PM",
      "term": "1490"
    },
    ... < more section objects >
  ]
}
```