

Select a definition | yourscope-api v1

yourscope-api

1.0 OAS3

https://localhost:7184/swagger/v1/swagger.json

Accour	nts	^
GET	/api/accounts/v1/check-registered/{email} Checks whether an email is already register or not.	ed to a user
Parameters	s T	ry it out
Name	Description	
email * req string (path)	A required path parameter representing an email address.	
Responses	S Company of the comp	
Code	Description	Links
200	Success	No links
	Media type text/plain Controls Accept header. Example Value Schema	
	true	
404	Not Found	No links

Media type

text/plain

Example Value Schema

{
 "type": "string",
 "title": "string",
 "status": 0,
 "detail": "string",
 "instance": "string",
 "additionalProp1": "string",
 "additionalProp2": "string",
 "additionalProp3": "string"
}

/api/accounts/v1/{email}/send-password-reset-Sends a password reset email to the email **GET** address specified if they are registered. email **Parameters** Try it out Name Description email * required A required path parameter representing an email address. string Responses Code Description Links 201 No links Created Media type text/plain Controls Accept header. Example Value Schema true 404 No links Not Found

text/plain

Example Value Schema

{
 "type": "string",
 "title": "string",
 "status": 0,
 "detail": "string",
 "instance": "string",
 "additionalProp1": "string",
 "additionalProp2": "string",
 "additionalProp3": "string"
}

/api/accounts/v1/{email}/send-password-reset-Sends a password reset email to the email **POST** address specified if they are registered. email **Parameters** Try it out Name Description email * required A required path parameter representing an email address. string Responses Code Description Links 201 No links Created Media type text/plain Controls Accept header. Example Value Schema true 404 No links Not Found

text/plain

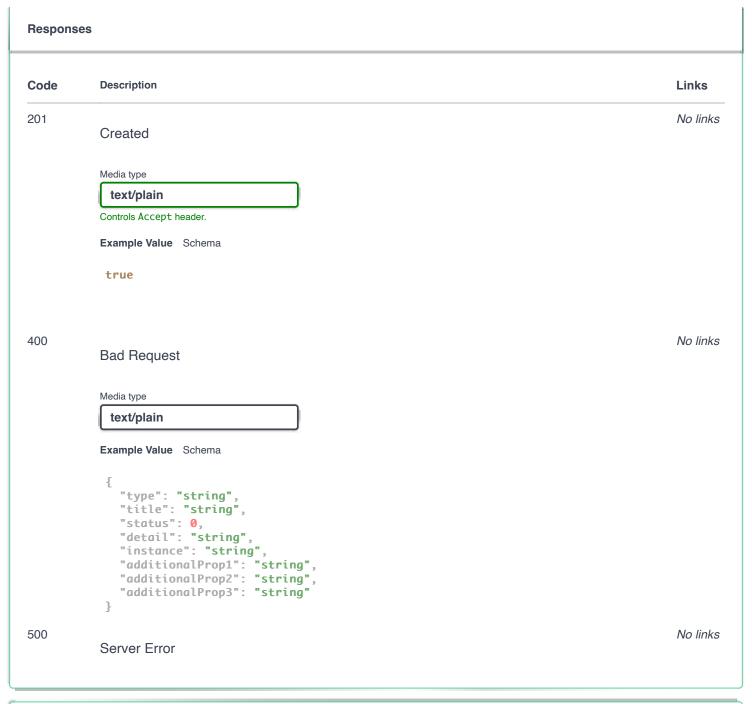
Example Value Schema

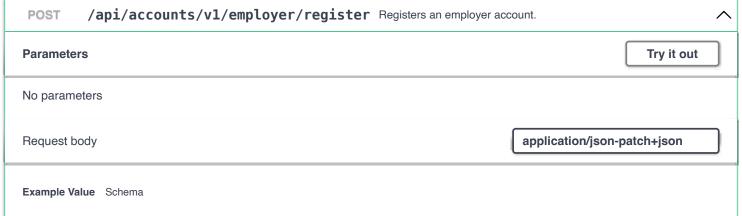
{
 "type": "string",
 "title": "string",
 "status": 0,
 "detail": "string",
 "instance": "string",
 "additionalProp1": "string",
 "additionalProp2": "string",
 "additionalProp3": "string",
}

```
/api/accounts/v1/student/register Registers a student account.
POST
Parameters
                                                                                                  Try it out
No parameters
                                                                              application/json-patch+json
Request body
Example Value Schema
   "email": "email@domain.com",
   "password": "Password1",
  "firstName": "Samuel",
  "middleName": "Leroy",
   "lastName": "Jackson",
   "birthday": "2023-08-04T22:45:55.274Z",
   "affiliation": "Wendat Village",
   "grade": 8
Responses
Code
           Description
                                                                                                      Links
201
                                                                                                      No links
           Created
           Media type
             text/plain
           Controls Accept header.
```

```
Example Value Schema
              true
400
                                                                                                                         No links
             Bad Request
             Media type
               text/plain
             Example Value Schema
                 "type": "string",
"title": "string",
                 "status": 0,
                 "detail": "string",
                 "instance": "string",
                 "additionalProp1": "string",
"additionalProp2": "string",
                 "additionalProp3": "string"
              }
500
                                                                                                                         No links
             Server Error
```

```
/api/accounts/v1/admin/register Registers an admin account.
POST
                                                                                                Try it out
Parameters
No parameters
Request body
                                                                             application/json-patch+json
Account details for admin registration
Example Value Schema
   "email": "email@domain.com",
   "password": "Password1",
   "firstName": "Samuel",
  "middleName": "Leroy",
   "lastName": "Jackson",
   "birthday": "2023-08-04T22:45:55.281Z",
   "affiliation": "Wendat Village",
   "grade": 8
```

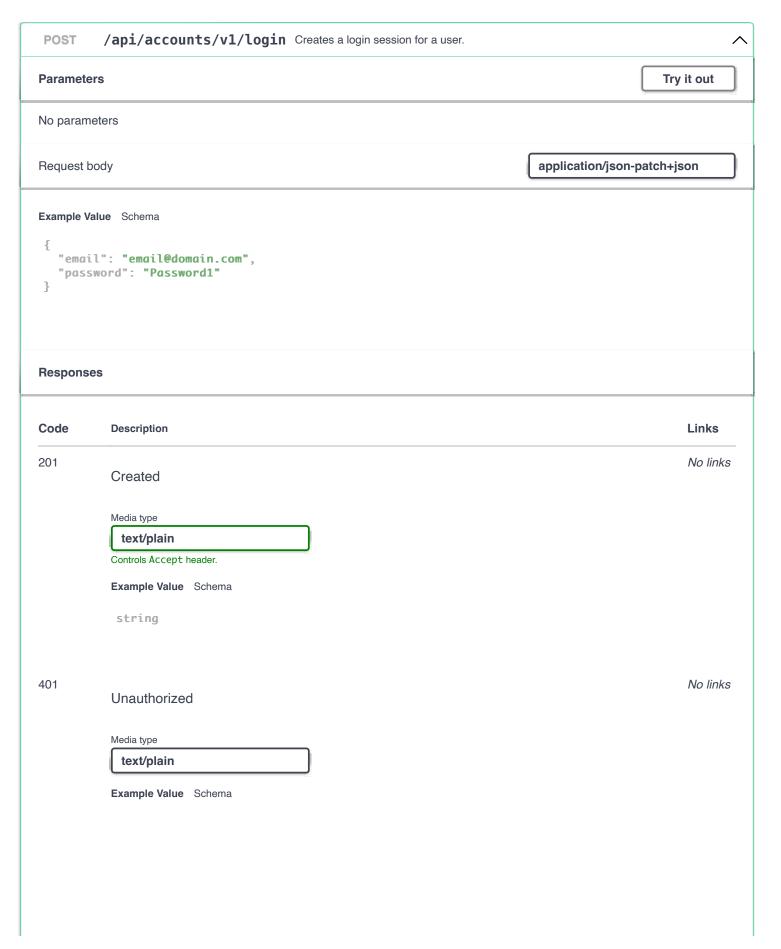




```
"email": "email@domain.com",
"password1",
"firstName": "Samuel",
"middleName": "Leroy",
"lastName": "Jackson",
"birthday": "2023-08-04T22:45:55.290Z",
"affiliation": "Wendat Village",
"grade": 8
}
```

Responses

Description Links Code 201 No links Created Media type text/plain Controls Accept header. Example Value Schema true No links 400 **Bad Request** Media type text/plain Example Value Schema "type": "string",
"title": "string", "status": 0,
"detail": "string",
"instance": "string", "additionalProp1": "string",
"additionalProp2": "string",
"additionalProp3": "string" } 500 No links Server Error

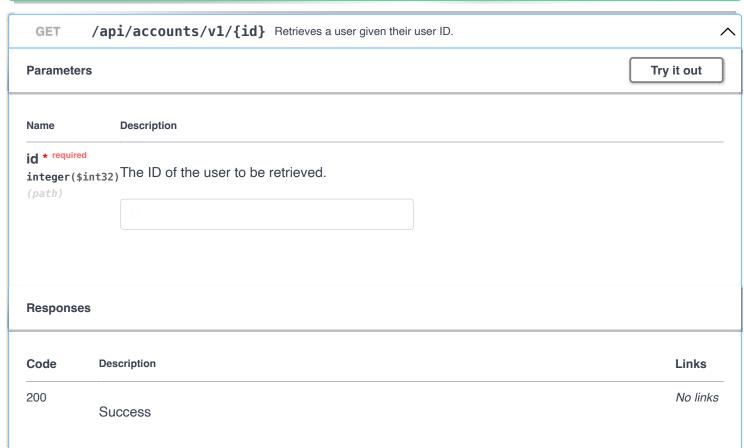


```
"type": "string",
    "title": "string",
    "status": 0,
    "detail": "string",
    "instance": "string",
    "additionalProp1": "string",
    "additionalProp2": "string",
    "additionalProp3": "string"
}

500

Server Error

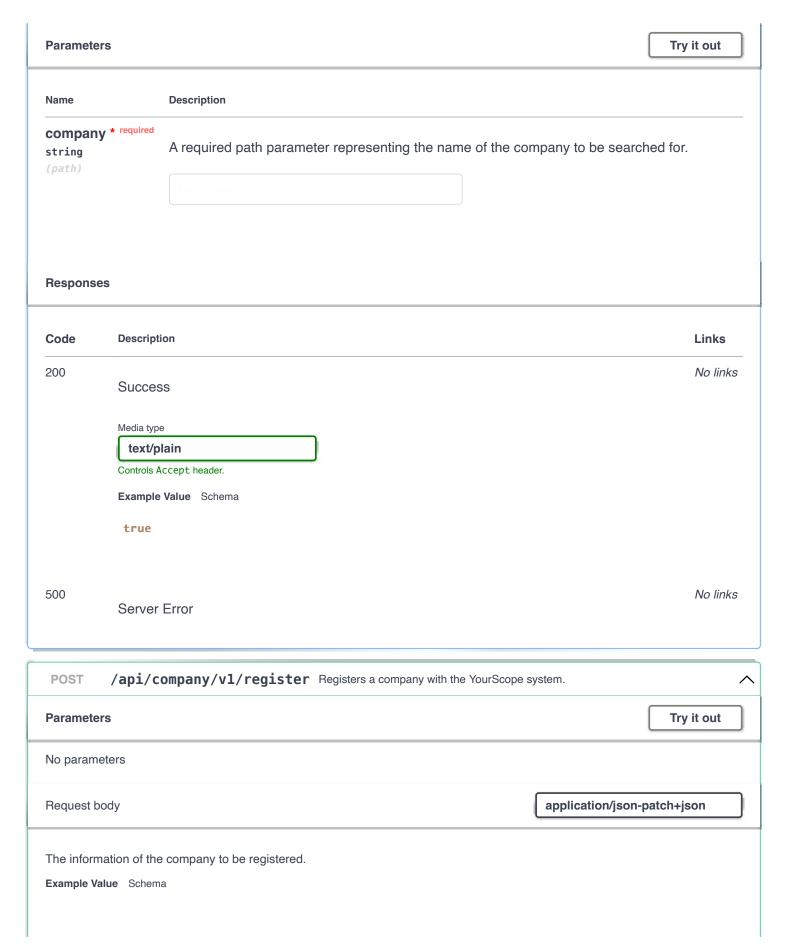
No links
```



Company

GET /api/company/v1/check-company-exist/{company}

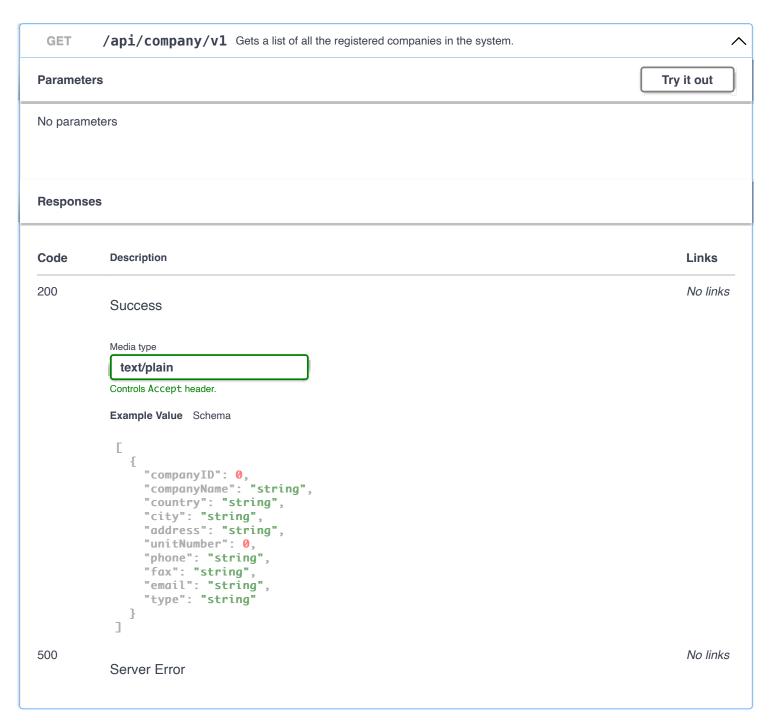
Used to check whether or not a company with a given name has been registered yet.



```
"companyID": 0,
"companyName": "string",
"country": "string",
"address": "string",
"unitNumber": 0,
"phone": "string",
"fax": "string",
"email": "string",
"type": "string"
```

Responses

Code Description Links No links 201 Created Media type text/plain Controls Accept header. Example Value Schema true 400 No links **Bad Request** Media type text/plain Example Value Schema "type": "string",
"title": "string", "status": 0,
"detail": "string",
"instance": "string", "additionalProp1": "string",
"additionalProp2": "string",
"additionalProp3": "string" } 500 No links Server Error



Events

POST /api/events/v1 Creates a new event and adds it into the system.

Parameters

No parameters

Try it out

Request body

application/json-patch+json

The details of the event to be created.

```
Example Value Schema
```

```
{
  "title": "string",
  "description": "string",
  "date": "2023-08-04T22:45:55.316Z",
  "location": "string",
  "userId": 0
```

Responses

Code Description Links

201 No links

Created

Media type

text/plain

Controls Accept header.

Example Value Schema

true

400 No links

Bad Request

Media type

text/plain

Example Value Schema

```
{
  "type": "string",
  "title": "string",
  "status": 0,
  "detail": "string",
  "instance": "string",
  "additionalProp1": "string",
  "additionalProp2": "string",
  "additionalProp3": "string"
}
```

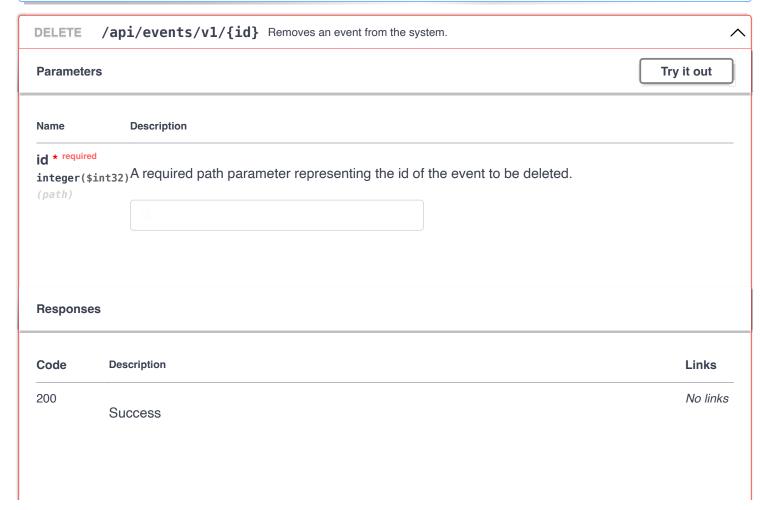
No links

500

Server Error

GET /a	pi/events/v1 Gets a list of events in the system based on filters provided as query parameters.
Parameters	Try it out
Name	Description
userld integer(\$int32 (query)	An optional query parameter (user's ID as an integer) used to filter the list of events by the user who created it.
<pre>schoolld integer(\$int32 (query)</pre>	An optional query parameter representing a school ID as an integer and used to filter the list of events by associated school.
offset integer(\$int32 (query)	An optional query parameter (integer) representing the number of records to skip (default: 0). Used for pagination.
<pre>count integer(\$int32 (query)</pre>	An optional query parameter (integer) representing the number of records to return (default: 10). Used for pagination.
Responses	
Code De	scription
200 Si	No links
_	text/plain
	ntrols Accept header.
Ex	ample Value Schema

```
Е
               "eventId": 0,
               "title": "string",
               "description": "string",
               "date": "2023-08-04T22:45:55.326Z",
               "location": "string",
               "userId": 0,
                "user": {
                 "userId": 0,
                 "email": "string",
                 "firstName": "string",
                 "middleName": "string",
                 "lastName": "string",
                 "birthday": "2023-08-04T22:45:55.326Z",
                 "role": 0,
                 "affiliation": "string",
                 "affiliationID": ∅,
                  "grade": 8
                "schoolId": 0,
                "school": {
                 "schoolId": 0,
                 "name": "string"
                 "address" · "strina"
                                                                                                No links
500
          Server Error
```

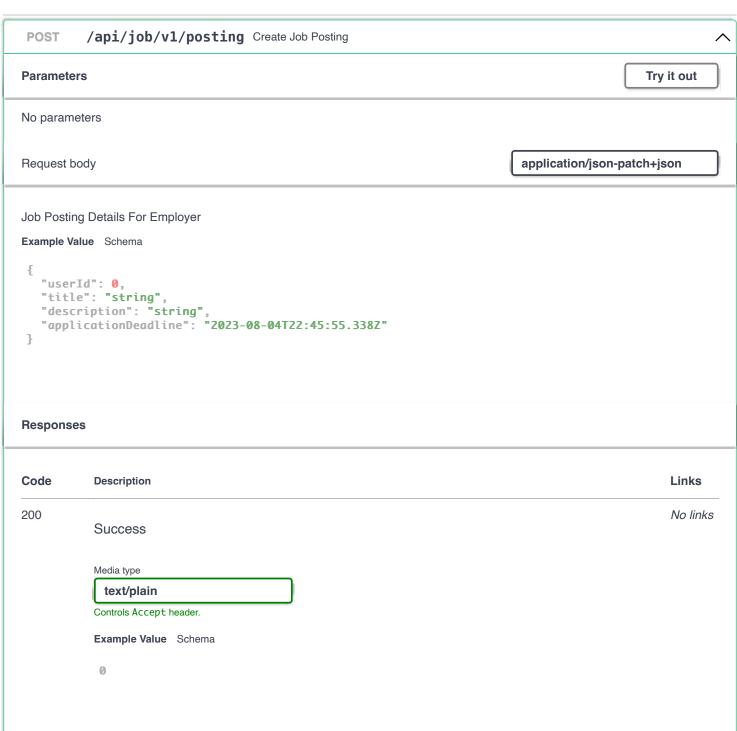




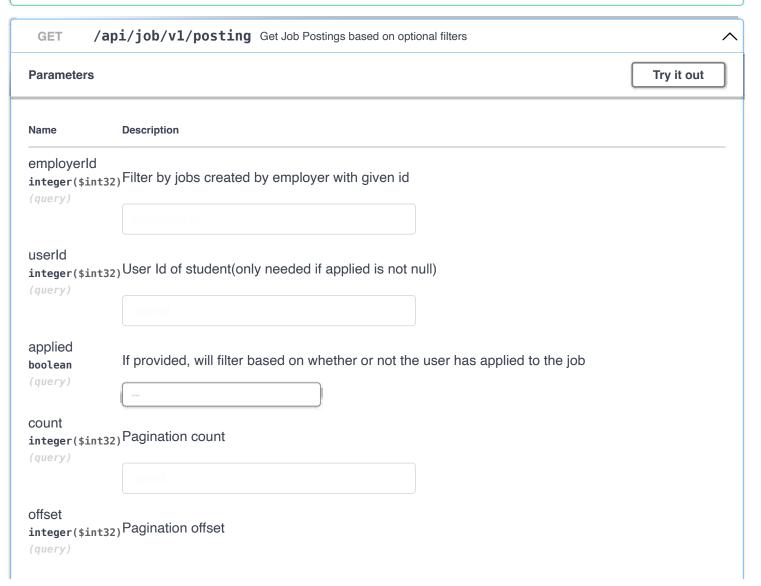
/api/events/v1/count Gets the number of events in the system based on filters provided as query parameters. **GET Parameters** Try it out Description Name userld integer(\$int32)An optional query parameter (user's ID as an integer) used to filter the list of events by the user who created it. schoolld integer(\$int32)An optional query parameter representing a school ID as an integer and used to filter the list of events by associated school. (query) Responses Code Description Links No links 200 Success Media type text/plain Controls Accept header. Example Value Schema

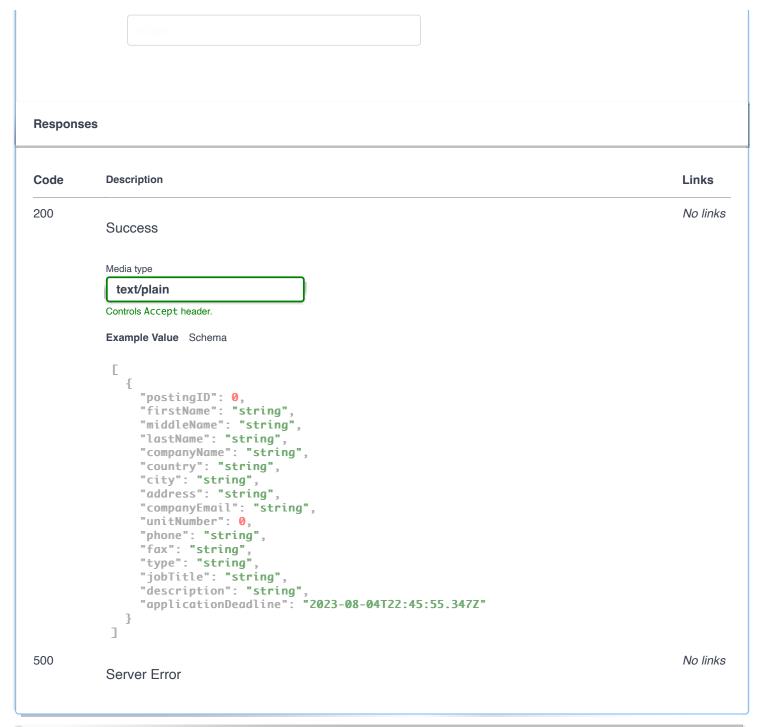
500 No links
Server Error

Job



400 **Bad Request** No links Media type text/plain Example Value Schema "type": "string", "title": "string", "status": 0, "detail": "string", "instance": "string", "additionalProp1": "string", "additionalProp2": "string",
"additionalProp3": "string" 500 No links Server Error

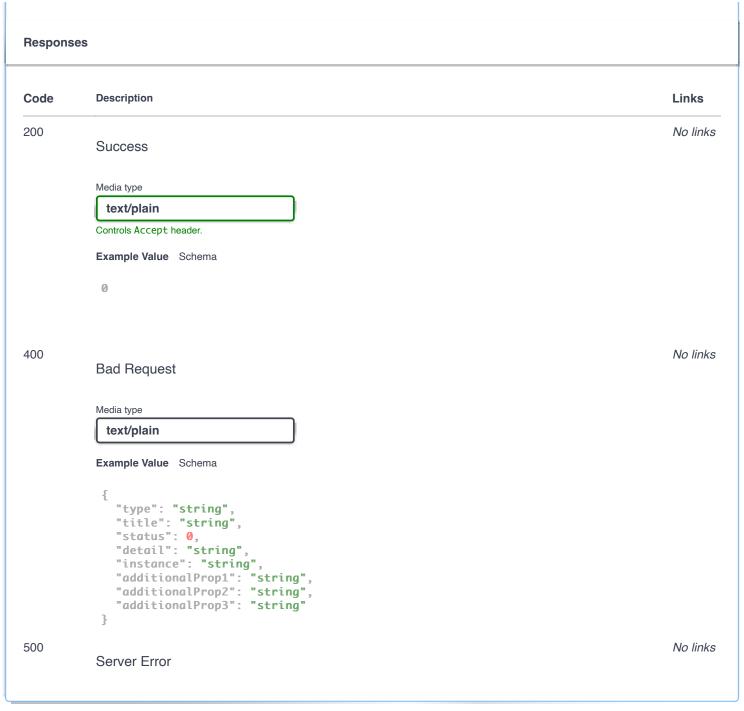






Respons	ees	
Code	Description	Links
200	Success	No links
	Media type text/plain Controls Accept header.	
	Example Value Schema Ø	
500	Server Error	No links

GET /ar	pi/job/v1/posting/count Get number of job postings based on optional filters		^
Parameters	Parameters Try it ou		
Name	Description		
employerld integer(\$int32 (query)	Filter by jobs created by employer with given id		
	User Id of student(only needed if applied is not null)		
(query)			
applied boolean (query)	If provided, will filter based on whether or not the user has applied to the job		



```
POST /api/job/v1/application Create Job Application

Parameters

No parameters

Request body

Application Create Job Application

Try it out

application/json-patch+json
```

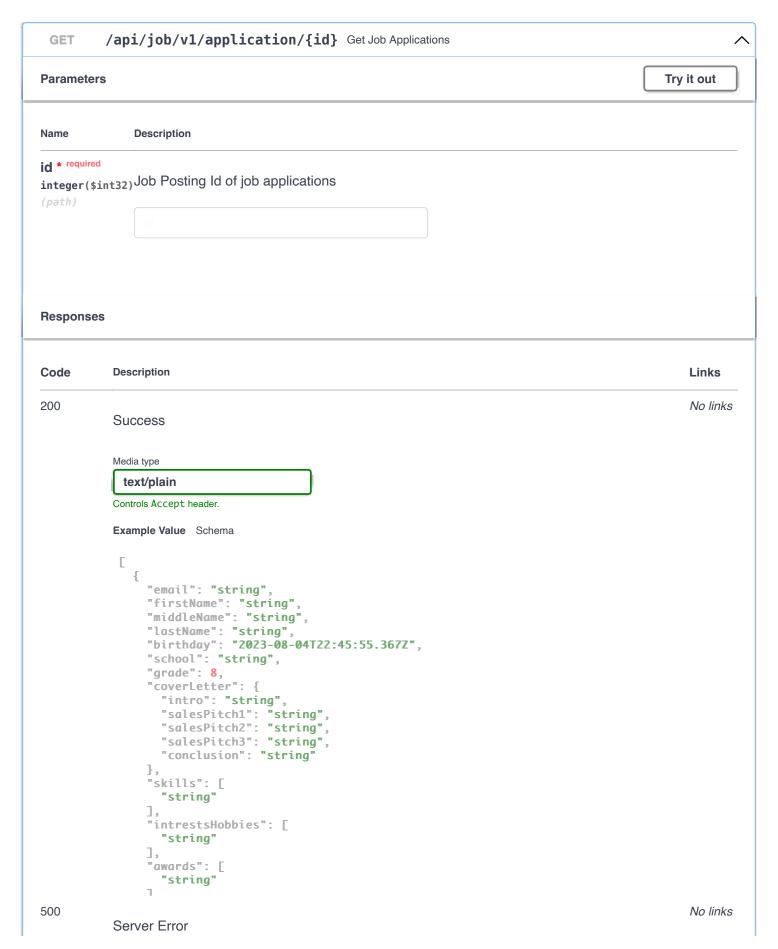
```
Job Application details

Example Value Schema

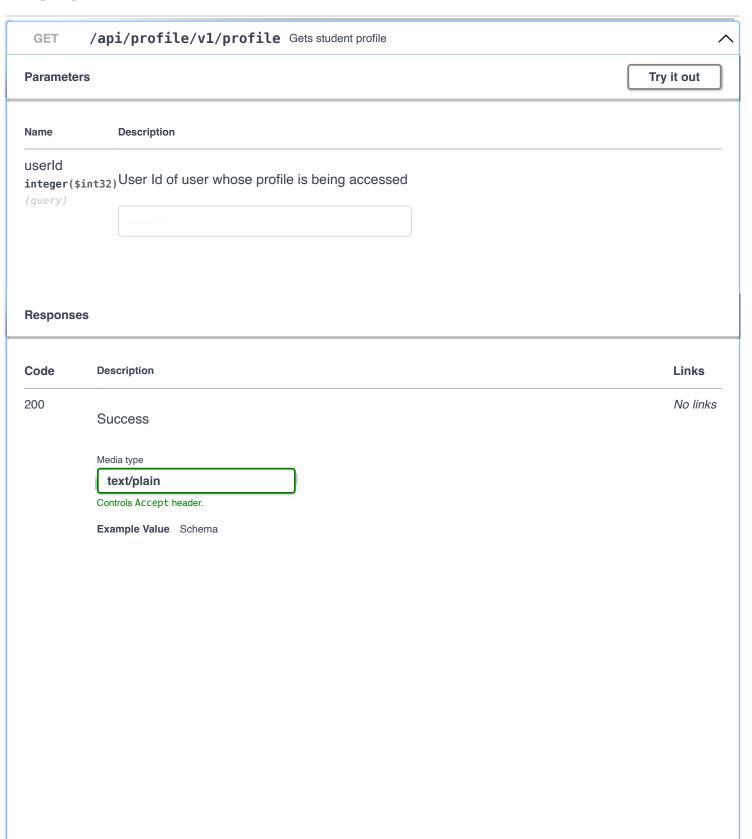
{
    "jobPostingId": 0,
    "userId": 0,
    "coverLetterId": 0
}
```

Responses

```
Code
            Description
                                                                                                                  Links
200
                                                                                                                  No links
            Success
            Media type
              text/plain
            Controls Accept header.
            Example Value Schema
             0
400
                                                                                                                  No links
            Bad Request
            Media type
              text/plain
            Example Value Schema
                "type": "string",
                "title": "string",
                "status": 0,
                "detail": "string",
                "instance": "string",
                "additionalProp1": "string",
"additionalProp2": "string",
                "additionalProp3": "string"
             }
500
                                                                                                                  No links
            Server Error
```

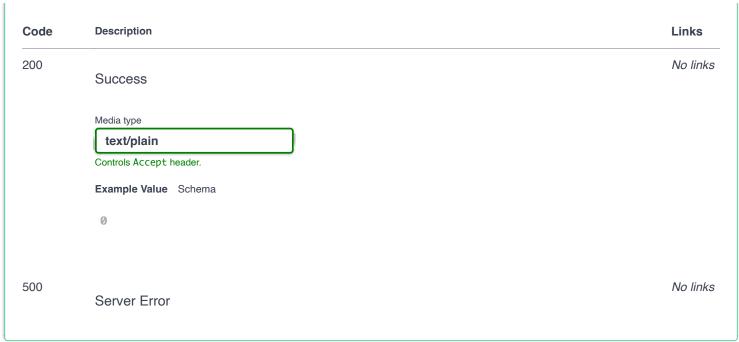


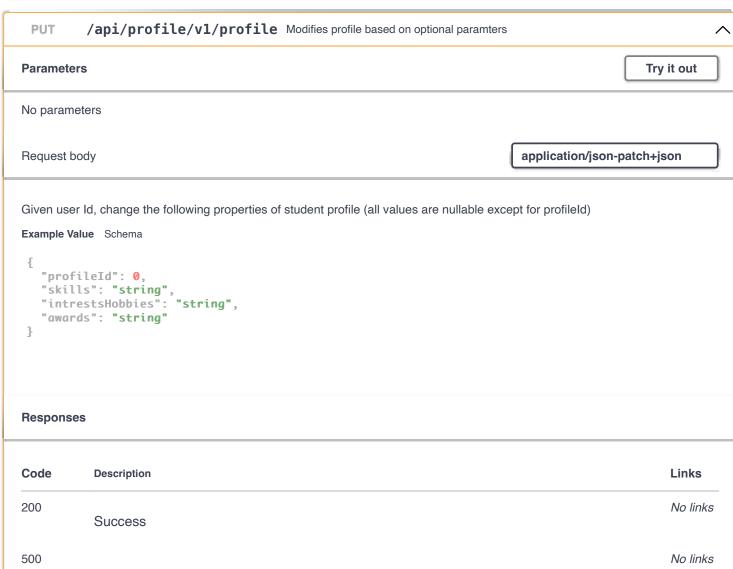
Profile



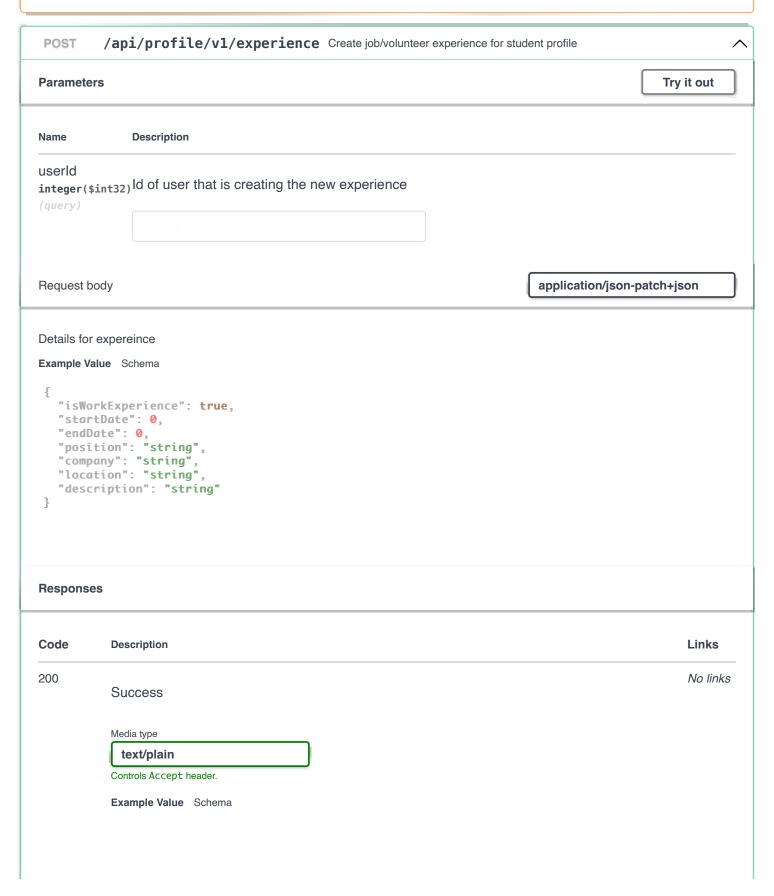
```
"profileId": 0,
              "user": {
                "userId": 0,
                "email": "string",
                "firstName": "string",
                "middleName": "string",
                "lastName": "string",
                "birthday": "2023-08-04T22:45:55.371Z",
                "role": 0,
                "affiliation": "string",
                "affiliationID": 0,
                "grade": 8
              "coverLetters": [
                   "intro": "string",
                  "salesPitch1": "string", "salesPitch2": "string",
                   "salesPitch3": "string",
                   "conclusion": "string"
              ],
              "experiences": [
204
                                                                                                      No links
           No Content
500
                                                                                                      No links
           Server Error
```







Server Error

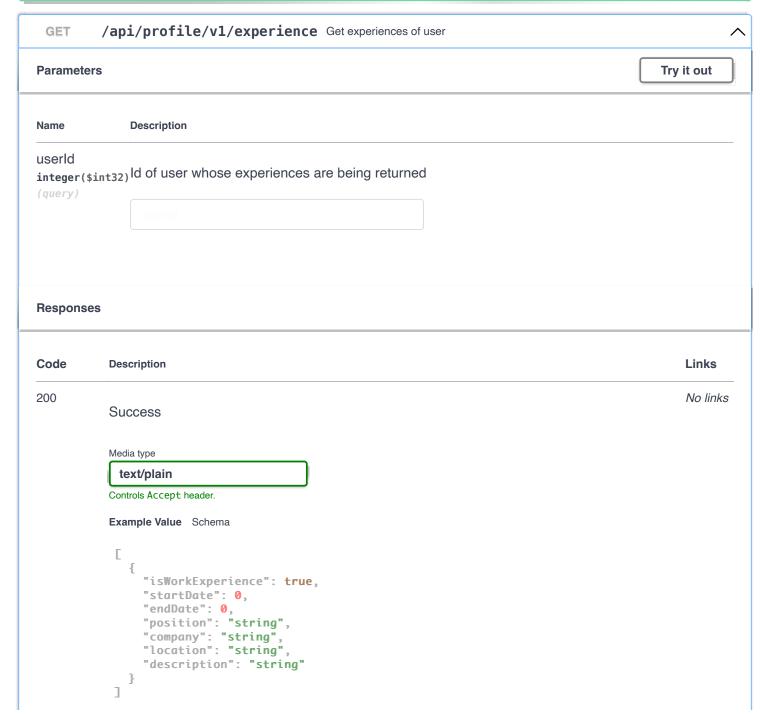


```
"isWorkExperience": true,
    "startDate": 0,
    "endDate": 0,
    "position": "string",
    "company": "string",
    "location": "string",
    "description": "string"
}

500

No links

Server Error
```



500

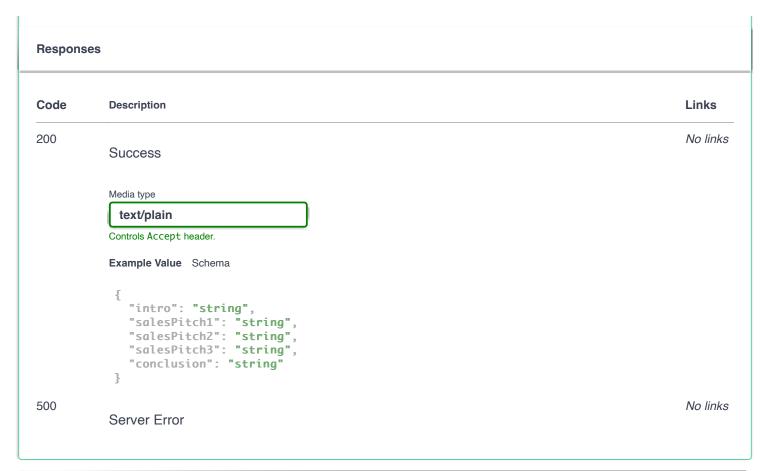
Server Error

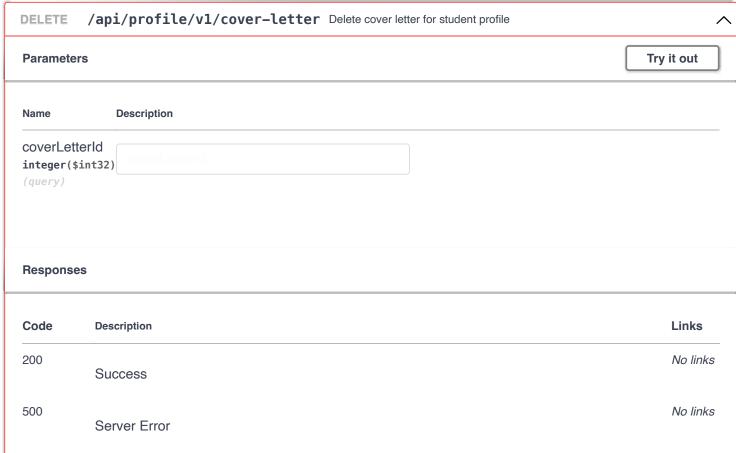
No links

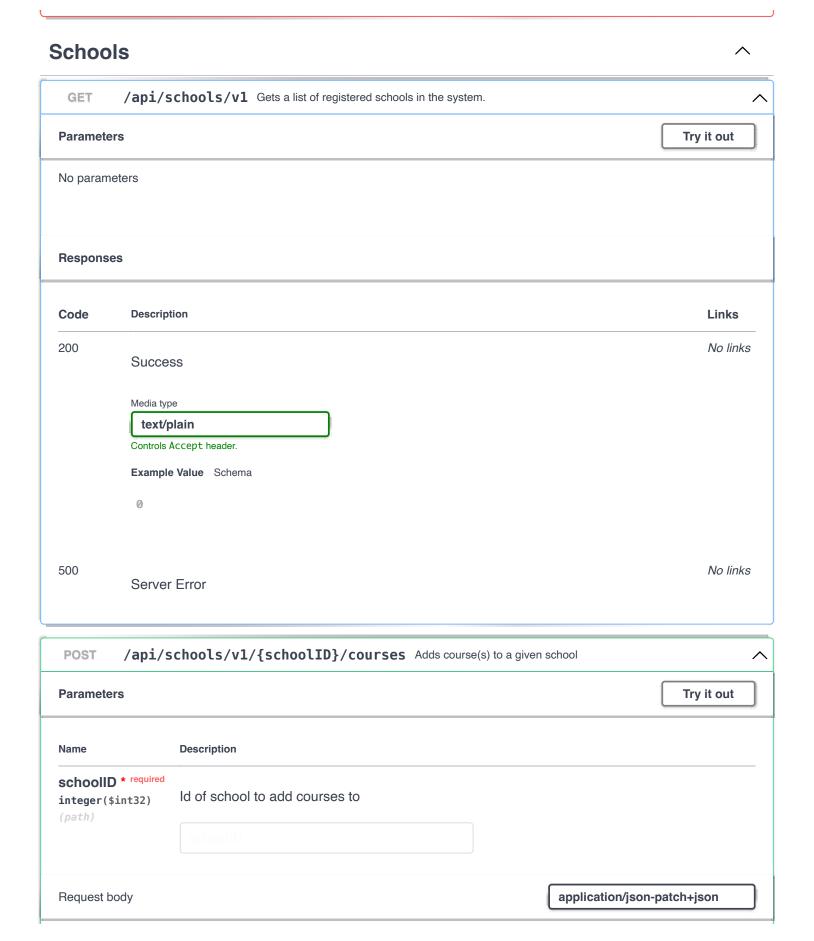
DELETE /api/profile/v1/experience Deletes experience based on experienceld	^	
Parameters	Try it out	
Name Description		
experienceId integer(\$int32) experienceId of experience to be deleted		
(query) experienceId		
Responses		
Code Description	Links	
200 Success	No links	
500 Server Error	No links	
GET /api/profile/v1/cover-letter Get Cover letters of user	^	
Parameters Try it out		
Name Description		
userId integer(\$int32) Id of user whose cover letters are being returned (query)		
Responses		

```
Code
                                                                                                                       Links
             Description
200
                                                                                                                       No links
             Success
             Media type
               text/plain
             Controls Accept header.
             Example Value Schema
              Е
                   "intro": "string",
                   "salesPitch1": "string", "salesPitch2": "string",
                   "salesPitch3": "string",
                   "conclusion": "string"
              ]
500
                                                                                                                       No links
             Server Error
```

```
/api/profile/v1/cover-letter Create cover letter for student profile
POST
Parameters
                                                                                                             Try it out
Name
                Description
userld
integer($int32) Id of user that is creating the new cover letter
(query)
Request body
                                                                                        application/json-patch+json
Details for cover letter
Example Value Schema
   "intro": "string",
   "salesPitch1": "string",
"salesPitch2": "string",
   "salesPitch3": "string",
   "conclusion": "string"
```







List of course(s) to add, if a course already exists, it uses the data that's already in the table.

Example Value Schema

```
"courseCode": "string",
    "name": "string",
    "description": "string",
    "discipline": "string",
    "type": "string",
    "grade": 0,
    "credits": 0,
    "prerequisites": "string"
}
```

Responses

Code	Description	Links
201	Created	No links
500	Server Error	No links

GET /api/schools/v1/courses/count Gets the number of filtered courses. **Parameters** Try it out Name Description integer(\$int32)An optional integer parameter representing the ID of a school used to filter courses by school. (query) searchQuery An optional string parameter used to filter courses by both course code and course title. string (query) Example: ICS2O ICS2O grade integer(\$int32) An optional integer between 9 and 12 inclusive used to filter courses by grade.

(query)	Example: 9	
	9	
disciplines string (query)	An optional comma seperated list of strings used to filter courses by discipline. Example: Math, Science, Geography Math, Science, Geography	
Responses		
Code	Description	Links
200	Success	No links

GET /aı	pi/schools/v1/courses Gets a filtered list of courses.
GE1 / G	J. Schools, VI, Courses and a microanical constant
Parameters	Try it out
Name	Description
schoolID integer(\$int32 (query)	An optional integer parameter representing the ID of a school used to filter courses by school.
searchQuery string (query)	An optional string parameter used to filter courses by both course code and course title.
(400.37)	Example: ICS20
	ICS2O
	2) An optional integer between 9 and 12 inclusive used to filter courses by grade.
(query)	Example: 9
	9
disciplines	

string (query)	An optional comma seperated list of strings used to filter courses by discipline. Example: Math, Science, Geography		
	Math,Science,Geography		
offset integer(\$int32) (query)	An optional query parameter (integer) representing the number of records to skip (default: 0). Used for pagination. Default value: 0		
	0		
<pre>count integer(\$int32) (query)</pre>	An optional query parameter (integer) representing the number of records to return (default: 10). Used for pagination.		
	Default value : 10		
	10		
Responses			
Code Des	cription		
200 Suc	200 No links Success		
DELETE /ap:	i/schools/v1/{schoolID}/courses/{courseID} Remove a course from a school. This will not remove the course from the Courses table.		
Parameters	Try it out		
Name	Description		
schoolID * requirements integer(\$int32) (path)	A required path parameter (integer) representing the ID of the school with the course to be removed from.		
	schoolID		
courseID * requirequinteger(\$int32)	A required path parameter (integer) representing the ID of the course to be removed from the		

(path)

school.



Student GET /api/student/v1/schedule/{studentID} Gets the schedule of a student given their user ID. **Parameters** Try it out Name Description studentID * required The user ID of the student whose schedule is to be retrieved. integer(\$int32) (path) Responses Description Links Code 200 No links Success

POST /api/student/v1/schedule/{studentID} Create a schedule for the a student given their user ID.

Parameters

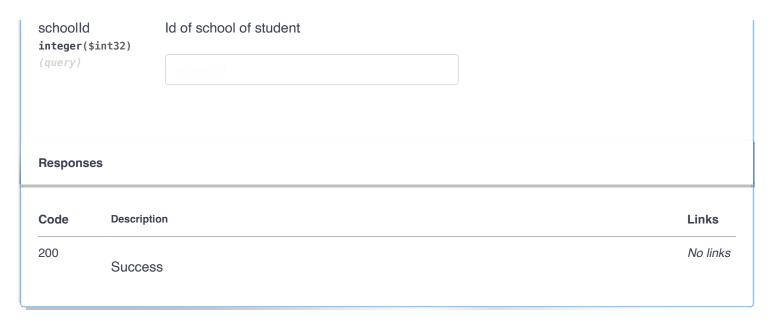
Try it out

Name Description

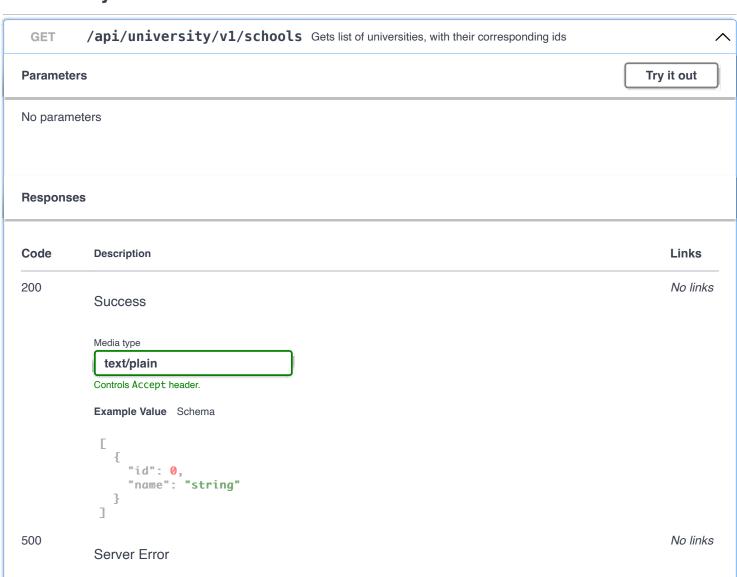
student integer((path)		The user ID of the student whose schedule is to be crea	ted.
Respons	ses Descriptio	on .	Links
200	Succes		No links
POST	/api/st	udent/v1/schedule/{studentID}/year/{year}	Adds a course to a student's schedule for

	<pre>tudent/v1/schedule/{studentID}/year/{year} e/{courseID}</pre>	Adds a course to a student's schedule for the given year.
Parameters		Try it out
Name	Description	
studentID * required integer(\$int32) (path)	The user ID of the student whose schedule will be added	d to.
year * ^{required} integer(\$int32) (path)	The year of study that the course should be added to.	
<pre>courseID * required integer(\$int32) (path)</pre>	The course ID of the course that will be added to the students	dent's schedule.
D		
Responses		
Code Descript	on	Links
200 Succes		No links

	tudent/v1/schedule/{studentID}/year/{year} e/{courseID}	Removes a course from a student's schedule given the student's user ID, the year of study, and the course ID to be removed.
Parameters		Try it out
Name	Description	
<pre>studentID * required integer(\$int32) (path)</pre>	The user ID of the student whose schedule will be remove	ved from.
year * required		
<pre>integer(\$int32) (path)</pre>	The year of study that the course should be removed from	m.
<pre>courseID * required integer(\$int32) (path)</pre>	The course ID of the course that will be removed from th	e student's schedule.
(pacis)		
Responses Code Descripti	ion	Links
		Al- E-L-
Succes	es	No links
GET /api/s	tudent/v1/insight/courses/{studentID} Getreco	comeneded courses for student
Parameters		Try it out
Name	Description	
<pre>studentID * required integer(\$int32) (path)</pre>	ld of student	
(been)		



University



GET /	api/university/v1/programs Gets list of univesity programs, based on optional filters	^
Parameters		Try it out
Name	Description	
UniversityId integer(\$int (query)	32)	
Search string (query)		
<pre>count integer(\$int (query)</pre>	Pagination count	
offset integer(\$int	Pagination offset	
(query)		
Responses		
Code I	Description	Links
200	Success	No links
1	Media type	
	text/plain	
Ċ	Controls Accept header.	
1	Example Value Schema	

```
"id": 0,
    "name": "string",
    "gradeRange": "string",
    "prerequisites": "string",
    "website": "string",
    "universityId": 0
}

500

Server Error

No links
```

Paramete	ers	Try it out
Name	Description	
Jniversit integer(\$ (query)		
Search string (query)		
		Linko
Code	es Description	Links No links
Response Code		Links No links
Code	Description	
Code	Description Success Media type text/plain	

Schemas Company companyID integer(\$int32) companyName string nullable: true country string nullable: true city string nullable: true address string nullable: true unitNumber integer(\$int32) nullable: true phone string nullable: true fax string nullable: true email string nullable: true type string nullable: true } Course courseCode string nullable: true string name nullable: true description string nullable: true discipline string nullable: true type string nullable: true grade integer(\$int32) credits Credits integer(\$int32) Enum: Array [5] prerequisites string nullable: true }

```
CoverLetter
   intro
                        string
                        nullable: true
   salesPitch1
                        string
                        nullable: true
   salesPitch2
                        string
                        nullable: true
   salesPitch3
                        string
                        nullable: true
   conclusion
                        string
                        nullable: true
}
Credits
           integer($int32)
Enum:
   Array [ 5 ]
Event
   eventId
                        integer($int32)
   title
                        string
                        nullable: true
   description
                        string
                        nullable: true
   date
                        string($date-time)
   location
                        string
                        nullable: true
   userId
                        integer($int32)
                        nullable: true
   user
                        User
                                {...}
                        integer($int32)
   schoolId
                        nullable: true
   school
                        School
                                  {...}
}
EventCreationDto
   title
                        string
                        nullable: true
                        string
   description
                        nullable: true
                        string($date-time)
   location
                        string
                        nullable: true
   userId
                        integer($int32)
}
```

```
Experience
   isWorkExperience
                         boolean
                         integer($int32)
integer($int32)
   startDate
   endDate
                         string
   position
                         nullable: true
   company
                         string
                         nullable: true
   location
                         string
                         nullable: true
   description
                         string
                         nullable: true
}
JobApplicationCreation
   jobPostingId
                         integer($int32)
                         integer($int32)
   userId
                         integer($int32)
   coverLetterId
                         nullable: true
}
JobApplicationDetails
   email
                         string
                         nullable: true
   firstName
                         string
                         nullable: true
   middleName
                         string
                         nullable: true
   lastName
                         string
                         nullable: true
                         string($date-time)
   birthday
                         nullable: true
   school
                         string
                         nullable: true
   grade
                         UserGrade integer($int32)
                             Array [ 6 ]
   coverLetter
                         CoverLetter
                                          {...}
   skills
                             [...]
   intrestsHobbies
                             [...]
   awards
                             [\ldots]
}
JobPostingCreation
   userId
                         integer($int32)
   title
                         string
                         nullable: true
   description
                         string
                         nullable: true
   applicationDeadline string($date-time)
}
```

```
JobPostingDetails
   postingID
                         integer($int32)
   firstName
                         string
                         nullable: true
                         string
   middleName
                         nullable: true
   lastName
                         string
                         nullable: true
   companyName
                         string
                         nullable: true
   country
                         string
                         nullable: true
   city
                         string
                         nullable: true
   address
                         string
                         nullable: true
   companyEmail
                         string
                         nullable: true
   unitNumber
                         integer($int32)
                         nullable: true
   phone
                         string
                         nullable: true
   fax
                         string
                         nullable: true
   type
                         string
                         nullable: true
   jobTitle
                         string
                         nullable: true
   description
                         string
                         nullable: true
   applicationDeadline string($date-time)
}
ProblemDetails
   type
                         string
                         nullable: true
   title
                         string
                         nullable: true
   status
                         integer($int32)
                         nullable: true
   detail
                         string
                         nullable: true
   instance
                         string
                         nullable: true
}
Profile
                         integer($int32)
   profileId
   user
                         User
                                 {...}
   coverLetters
                            [...]
   experiences
                            [...]
   skills
                         string
                         nullable: true
   intrestsHobbies
                         string
                         nullable: true
   awards
                         string
                         nullable: true
}
```

```
ProfileCreation
   userId
                        integer($int32)
   skills
                        string
                        nullable: true
   intrestsHobbies
                        string
                        nullable: true
   awards
                        string
                        nullable: true
}
ProfileDetails
   profileId
                        integer($int32)
   skills
                         string
                        nullable: true
   intrestsHobbies
                         string
                        nullable: true
   awards
                        string
                        nullable: true
}
School
   schoolId
                        integer($int32)
   name
                         string
                        nullable: true
   address
                         string
                        nullable: true
   schoolCourses
                            [...]
}
SchoolCourse
   schoolId
                        integer($int32)
   courseId
                        integer($int32)
   school
                        School
                                    {...}
   course
                        Course
                                    {...}
}
UniProgram
                {
   id
                        integer($int32)
   name
                         string
                        nullable: true
   gradeRange
                         string
                        nullable: true
   language
                         string
                        nullable: true
   prerequisites
                        string
                        nullable: true
   website
                        string
                        nullable: true
   universityId
                        integer($int32)
}
```

```
University
   id
                        integer($int32)
   name
                        string
                        nullable: true
}
User
   userId
                        integer($int32)
   email
                        string
                        nullable: true
   firstName
                        string
                        nullable: true
   middleName
                        string
                        nullable: true
   lastName
                        string
                        nullable: true
   birthday
                        string($date-time)
                        nullable: true
   role
                        UserRole integer($int32)
                            Array [ 3 ]
   affiliation
                        string
                        nullable: true
   affiliationID
                        integer($int32)
                        nullable: true
   grade
                        UserGrade integer($int32)
                        Enum:
                            Array [ 6 ]
}
UserGrade
               integer($int32)
Enum:
   Array [ 6 ]
UserLoginDto
   email
                        string
                        nullable: true
                        example: email@domain.com
                        Email: A required string representing the user's email address.
   password
                        string
                        nullable: true
                        example: Password1
                        A string required representing the user's password.
```

}

```
UserRegistrationDto
                         {
   email
                        string
                        nullable: true
                        example: email@domain.com
                        Email: A required string representing the email address of the user used for
                        registration.
   password
                        string
                        nullable: true
                        example: Password1
                        Password: A required string storing the user's password used for registration.
   firstName
                        strina
                        nullable: true
                        example: Samuel
                        First Name: A required string representing the user's first name.
   middleName
                        string
                        nullable: true
                        example: Leroy
                        Middle Name: An optional string representing the user's middle name.
   lastName
                        string
                        nullable: true
                        example: Jackson
                        Last Name: A required string representing the user's last name.
   birthday
                        string($date-time)
                        nullable: true
                        Birthday: An optional date string representing the user's birthday.
   affiliation
                        string
                        nullable: true
                        example: Wendat Village
                        Affiliation: A required string representing either the name of the company the
                        user works at (employers) or the name of the school the user is associated with
                        (students and admins).
   grade
                        UserGrade integer($int32)
                        Enum:
                            Array [ 6 ]
}
UserRole
             integer($int32)
Enum:
   Array [ 3 ]
```