



Select a definition

yourscope-api v1

yourscope-api1.0OAS3

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

Accounts

GET

/api/accounts/v1/check-registered/{email}

Checks whether an email is already registered to a user or not.

Parameters

Try it out

Name	Description
email * required string (path)	A required path parameter representing an email address.

email

Responses

Code	Description	Links
200	<div>Success</div> <div>Media type</div> <div>text/plain</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div>true</div>	No links
404	<div>Not Found</div>	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"
}
```

GET

/api/accounts/v1/{email}/send-password-reset-email

Sends a password reset email to the email address specified if they are registered.

Parameters

Try it out

Name	Description
email * required string (path)	A required path parameter representing an email address. <div><input type="text" value="email"/></div>

Responses

Code	Description	Links
201	Created <div><p>Media type</p><p>text/plain</p><p>Controls Accept header.</p><p>Example Value Schema</p><p>true</p></div>	No links
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"
}
```

POST

/api/accounts/v1/{email}/send-password-reset-email

Sends a password reset email to the email address specified if they are registered.

Parameters

Try it out

Name	Description
email * required string (path)	A required path parameter representing an email address. <div>email</div>

Responses

Code	Description	Links
201	Created <div><p>Media type</p><p>text/plain</p><p>Controls Accept header.</p><p>Example Value Schema</p><p>true</p></div>	No links
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"
}
```

POST **/api/accounts/v1/student/register** Registers a student account.

Parameters

Try it out

No parameters

Request body

application/json-patch+json

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	Created	No links

Media type

text/plain

Controls Accept header.

Example Value

Schema

true

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

Server Error

No links

POST

/api/accounts/v1/admin/register

Registers an admin account.

Parameters

Try it out

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
}
```

Responses

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

POST

/api/accounts/v1/employer/register

Registers an employer account.

Parameters

Try it out

No parameters

Request body

application/json-patch+json

Example Value

Schema

```
{
  "email": "email@domain.com",
  "password": "Password1",
  "firstName": "Samuel",
  "middleName": "Leroy",
  "lastName": "Jackson",
  "birthday": "2023-08-04T22:45:55.290Z",
  "affiliation": "Wendat Village",
  "grade": 8
}
```

Responses

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

POST

/api/accounts/v1/login

Creates a login session for a user.

Parameters

Try it out

No parameters

Request body

application/json-patch+json

Example Value

Schema

```
{
  "email": "email@domain.com",
  "password": "Password1"
}
```

Responses

Code	Description	Links
201	Created	No links
	<div>Media type</div> <div>text/plain</div> <div>Controls Accept header.</div> <div>Example Value</div> <div>Schema</div> <div>string</div>	
401	Unauthorized	No links
	<div>Media type</div> <div>text/plain</div> <div>Example Value</div> <div>Schema</div>	


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

500

Server Error

No links

GET

/api/accounts/v1/{id}

Retrieves a user given their user ID.

^

Parameters

Try it out

Name	Description
<div><div>id ★ required</div><div>integer(\$int32)</div><div>(path)</div></div>	The ID of the user to be retrieved.

id

Responses

Code	Description	Links
200	Success	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.

^

Parameters

Try it out

Name	Description
company * required string (path)	A required path parameter representing the name of the company to be searched for.
	<div>company</div>

Responses

Code	Description	Links
200	Success Media type <div>text/plain</div> Controls Accept header. Example Value Schema true	No links
500	Server Error	No links

POST /api/company/v1/register Registers a company with the YourScope system.



Parameters

Try it out

No parameters

Request body

application/json-patch+json

The information of the company to be registered.

Example Value Schema

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

Responses

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

GET

/api/company/v1

Gets a list of all the registered companies in the system.

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	<div>Success</div> <div>Media type</div> <div>text/plain</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div>[{ "companyID": 0, "companyName": "string", "country": "string", "city": "string", "address": "string", "unitNumber": 0, "phone": "string", "fax": "string", "email": "string", "type": "string" }]</div>	No links
500	<div>Server Error</div>	No links

Events

POST

/api/events/v1

Creates a new event and adds it into the system.

Parameters

Try it out

No parameters

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	<div>Created</div> <div>Media type</div> <div>text/plain</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div>true</div>	No links
400	<div>Bad Request</div> <div>Media type</div> <div>text/plain</div> <div>Example Value Schema</div> <div>{ "type": "string", "title": "string", "status": 0, "detail": "string", "instance": "string", "additionalProp1": "string", "additionalProp2": "string", "additionalProp3": "string" }</div>	No links
500		No links

Server Error

GET

/api/events/v1

Gets a list of events in the system based on filters provided as query parameters.

Parameters

Try it out

Name	Description
userId 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.
	<div>userId</div>
schoolId integer(\$int32) (query)	An optional query parameter representing a school ID as an integer and used to filter the list of events by associated school.
	<div>schoolId</div>
offset integer(\$int32) (query)	An optional query parameter (integer) representing the number of records to skip (default: 0). Used for pagination.
	<div>offset</div>
count integer(\$int32) (query)	An optional query parameter (integer) representing the number of records to return (default: 10). Used for pagination.
	<div>count</div>

Responses

Code	Description	Links
200	Success	No links
	Media type	
	<div>text/plain</div>	
	Controls Accept header.	
	Example Value	Schema

```
[
  {
    "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": 0,
      "grade": 8
    },
    "schoolId": 0,
    "school": {
      "schoolId": 0,
      "name": "string",
      "address": "string"
    }
  }
]
```

500

Server Error

No links

DELETE

/api/events/v1/{id}

Removes an event from the system.

⌵

Parameters

Try it out

Name	Description
id ★ required	
integer(\$int32)	A required path parameter representing the id of the event to be deleted.
(path)	
<input type="text" value="id"/>	

Responses

Code	Description	Links
200	Success	No links

Media type

text/plain

Controls Accept header.

Example Value

Schema

true

500

Server Error

No links

GET

/api/events/v1/count

Gets the number of events in the system based on filters provided as query parameters.

Parameters

Try it out

Name	Description
userId 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.
<div>userId</div>	
schoolId integer(\$int32) (query)	An optional query parameter representing a school ID as an integer and used to filter the list of events by associated school.
<div>schoolId</div>	

Responses

Code	Description	Links
200	Success	No links

Media type

text/plain

Controls Accept header.

Example Value

Schema

0

500

Server Error

No links

Job

^

POST

/api/job/v1/posting

Create Job Posting

^

Parameters

Try it out

No parameters

Request body

application/json-patch+json

Job Posting Details For Employer

Example Value

Schema

```
{  "userId": 0,  "title": "string",  "description": "string",  "applicationDeadline": "2023-08-04T22:45:55.338Z"}
```

Responses

Code	Description	Links
200	Success	No links

Media type

text/plain

Controls Accept header.

Example Value

Schema

0

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

Server Error

No links

GET

/api/job/v1/posting

Get Job Postings based on optional filters

Parameters

Try it out

Name	Description
employerId integer(\$int32) (query)	Filter by jobs created by employer with given id
userId integer(\$int32) (query)	User Id of student(only needed if applied is not null)
applied boolean (query)	If provided, will filter based on whether or not the user has applied to the job
count integer(\$int32) (query)	Pagination count
offset integer(\$int32) (query)	Pagination offset

alltest

Responses

Code	Description	Links
200	<div>Success</div> <div>Media type</div> <div>text/plain</div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div>[{ "postingID": 0, "firstName": "string", "middleName": "string", "lastName": "string", "companyName": "string", "country": "string", "city": "string", "address": "string", "companyEmail": "string", "unitNumber": 0, "phone": "string", "fax": "string", "type": "string", "jobTitle": "string", "description": "string", "applicationDeadline": "2023-08-04T22:45:55.347Z" }]</div>	No links
500	<div>Server Error</div>	No links

DELETE /api/job/v1/posting/{id} Deletes job posting

Parameters

Try it out

Name	Description
id * required	
integer(\$int32)	Id of job posting to delete
(path)	

Responses

Code	Description	Links
200	<div>Success</div> <div><div>Media type</div><div>text/plain</div><div>Controls Accept header.</div><div><div>Example Value</div><div>Schema</div><div>0</div></div></div>	No links
500	<div>Server Error</div>	No links

GET

/api/job/v1/posting/count

Get number of job postings based on optional filters



Parameters

Try it out

Name	Description
<div>employerId</div> <div>integer(\$int32)</div> <div>(query)</div>	<div>Filter by jobs created by employer with given id</div> <div><div>employerId</div></div>
<div>userId</div> <div>integer(\$int32)</div> <div>(query)</div>	<div>User Id of student(only needed if applied is not null)</div> <div><div>userId</div></div>
<div>applied</div> <div>boolean</div> <div>(query)</div>	<div>If provided, will filter based on whether or not the user has applied to the job</div> <div><div></div></div>

Responses

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

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

Parameters

Try it out

No parameters

Request body

application/json-patch+json

Job Application details

Example Value Schema

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

Responses

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

GET

/api/job/v1/application/{id}

Get Job Applications

⌵

Parameters

Try it out

Name	Description
id ★ required	Job Posting Id of job applications
integer(\$int32) (path)	<div><div></div></div>

Responses

Code	Description	Links
200	<div>Success</div> <div>Media type</div> <div><div>text/plain</div></div> <div>Controls Accept header.</div> <div>Example Value Schema</div> <div><pre>[{ "email": "string", "firstName": "string", "middleName": "string", "lastName": "string", "birthday": "2023-08-04T22:45:55.367Z", "school": "string", "grade": 8, "coverLetter": { "intro": "string", "salesPitch1": "string", "salesPitch2": "string", "salesPitch3": "string", "conclusion": "string" }, "skills": ["string"], "intrestsHobbies": ["string"], "awards": ["string"] }]</pre></div>	No links
500	Server Error	No links

Profile

^

GET

/api/profile/v1/profile

Gets student profile

^

Parameters

Try it out

Name	Description
userId	User Id of user whose profile is being accessed
<div>integer(\$int32) (query)</div>	
<div>userId</div>	

Responses

Code	Description	Links
200	Success	No links
	<div>Media type</div> <div>text/plain</div> <div>Controls Accept header.</div> <div>Example Value Schema</div>	


```
{
  "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 Content	No links
500	Server Error	No links

POST

/api/profile/v1/profile

Create new profile for user

⌵

Parameters

Try it out

No parameters

Request body

application/json-patch+json

Details of new profile

Example ValueSchema

```
{
  "userId": 0,
  "skills": "string",
  "intrestsHobbies": "string",
  "awards": "string"
}
```

Responses

Code	Description	Links
200	Success	No links
	<div>Media type</div> <div>text/plain</div> <div>Controls Accept header.</div> <div>Example ValueSchema</div> <div>0</div>	
500	Server Error	No links

PUT

/api/profile/v1/profile

Modifies profile based on optional paramters

Parameters

Try it out

No parameters

Request body

application/json-patch+json

Given user Id, change the following properties of student profile (all values are nullable except for profileId)

Example ValueSchema

```
{  "profileId": 0,  "skills": "string",  "intrestsHobbies": "string",  "awards": "string"}
```

Responses

Code	Description	Links
200	Success	No links
500		No links

Server Error

POST

/api/profile/v1/experience

Create job/volunteer experience for student profile

Parameters

Try it out

Name	Description
userId	
integer(\$int32)	Id of user that is creating the new experience
(query)	

userId

Request body

application/json-patch+json

Details for expereince

Example Value Schema

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

Responses

Code	Description	Links
200	Success	No links

Media type

text/plain

Controls Accept header.

Example Value Schema

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

500

Server Error

No links

GET

/api/profile/v1/experience

Get experiences of user

^

Parameters

Try it out

Name	Description
userId	
integer(\$int32)	Id of user whose experiences are being returned
(query)	

Responses

Code	Description	Links
200	Success	No links

Media type

text/plain

Controls Accept header.

Example Value

Schema

[
 {
 "isWorkExperience": true,
 "startDate": 0,
 "endDate": 0,
 "position": "string",
 "company": "string",
 "location": "string",
 "description": "string"
 }
]

500

Server Error

No links

DELETE

/api/profile/v1/experience

Deletes experience based on experienceId

⌵

Parameters

Try it out

Name	Description
experienceId	experienceId of experience to be deleted

integer(\$int32)

(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	Id of user whose cover letters are being returned

integer(\$int32)

(query)

userId

Responses

Code	Description	Links
200	<div>Success</div> <div>Media type</div> <div>text/plain</div> <div>Controls Accept header.</div> <div>Example ValueSchema</div> <div>[{ "intro": "string", "salesPitch1": "string", "salesPitch2": "string", "salesPitch3": "string", "conclusion": "string" }]</div>	No links
500	<div>Server Error</div>	No links

POST

/api/profile/v1/cover-letter

Create cover letter for student profile

Parameters

Try it out

Name	Description
userId	
integer(\$int32)	Id of user that is creating the new cover letter
(query)	

userId

Request body

application/json-patch+json

Details for cover letter

Example ValueSchema

{
 "intro": "string",
 "salesPitch1": "string",
 "salesPitch2": "string",
 "salesPitch3": "string",
 "conclusion": "string"
}

Responses

Code	Description	Links
200	<div>Success</div> <div>Media type</div> <div><div>text/plain</div></div> <div>Controls Accept header.</div> <div><div>Example Value</div><div>Schema</div><div><pre>{ "intro": "string", "salesPitch1": "string", "salesPitch2": "string", "salesPitch3": "string", "conclusion": "string" }</pre></div></div>	No links
500	<div>Server Error</div>	No links

DELETE /api/profile/v1/cover-letter Delete cover letter for student profile

Parameters

Try it out

Name	Description
coverLetterId integer(\$int32) (query)	<div>coverLetterId</div>

Responses

Code	Description	Links
200	Success	No links
500	Server Error	No links

Schools



GET

/api/schools/v1

Gets a list of registered schools in the system.

⌵

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	<div>Success</div> <div>Media type</div> <div>text/plain</div> <div>Controls Accept header.</div> <div>Example ValueSchema</div> <div>0</div>	No links
500	<div>Server Error</div>	No links

POST

/api/schools/v1/{schoolID}/courses

Adds course(s) to a given school

⌵

Parameters

Try it out

Name	Description
<div>schoolID ★ required</div> <div>integer(\$int32)</div> <div>(path)</div>	<div>Id of school to add courses to</div> <div>schoolID</div>

Request body

application/json-patch+json

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
schoolID	
integer(\$int32)	An optional integer parameter representing the ID of a school used to filter courses by school.
(query)	
	<input type="text" value="schoolID"/>
searchQuery	
string	An optional string parameter used to filter courses by both course code and course title.
(query)	Example : ICS2O
	<input type="text" value="ICS2O"/>
grade	
integer(\$int32)	An optional integer between 9 and 12 inclusive used to filter courses by grade.

*(query)**Example : 9***disciplines**
string*(query)*

An optional comma seperated list of strings used to filter courses by discipline.

*Example : Math,Science,Geography***Responses**

Code	Description	Links
200	Success	No links

GET **/api/schools/v1/courses** Gets a filtered list of courses.**Parameters**[Try it out](#)

Name	Description
schoolID	
integer(\$int32)	An optional integer parameter representing the ID of a school used to filter courses by school.
<i>(query)</i>	
	<input type="text" value="schoolID"/>
searchQuery	
string	An optional string parameter used to filter courses by both course code and course title.
<i>(query)</i>	<i>Example : ICS20</i>
	<input type="text" value="ICS20"/>
grade	
integer(\$int32)	An optional integer between 9 and 12 inclusive used to filter courses by grade.
<i>(query)</i>	<i>Example : 9</i>
	<input type="text" value="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

count
integer(\$int32)
(query)

An optional query parameter (integer) representing the number of records to return (default: 10). Used for pagination.
Default value : 10

10

Responses

Code	Description	Links
200	Success	No links

DELETE

/api/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 * required integer(\$int32) <i>(path)</i>	A required path parameter (integer) representing the ID of the school with the course to be removed from.
	<div>schoolID</div>
courseID * required integer(\$int32) <i>(path)</i>	A required path parameter (integer) representing the ID of the course to be removed from the school.

courseID

Responses

Code	Description	Links
200	Success	No links

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 integer(\$int32) (path)	The user ID of the student whose schedule is to be retrieved.
studentID	

Responses

Code	Description	Links
200	Success	No links

POST

/api/student/v1/schedule/{studentID}

Create a schedule for the 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 created.

integer(\$int32)

(path)

studentID

Responses

Code	Description	Links
200	Success	No links

POST

/api/student/v1/schedule/{studentID}/year/{year}/course/{courseID}

Adds a course to a student's schedule for the given year.

Parameters

Try it out

Name	Description
<div><div><div><div>studentID</div><div>*</div><div>required</div></div><div></div><div>The user ID of the student whose schedule will be added to.</div></div><div><div>integer(\$int32)</div><div>(path)</div><div><div>studentID</div></div></div></div>	
<div><div><div><div>year</div><div>*</div><div>required</div></div><div></div><div>The year of study that the course should be added to.</div></div><div><div>integer(\$int32)</div><div>(path)</div><div><div>year</div></div></div></div>	
<div><div><div><div>courseID</div><div>*</div><div>required</div></div><div></div><div>The course ID of the course that will be added to the student's schedule.</div></div><div><div>integer(\$int32)</div><div>(path)</div><div><div>courseID</div></div></div></div>	

Responses

Code	Description	Links
200	Success	No links

DELETE

/api/student/v1/schedule/{studentID}/year/{year}/course/{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
<div><div>studentID * required</div><div>integer(\$int32)</div><div>(path)</div></div>	<div>The user ID of the student whose schedule will be removed from.</div> <div>studentID</div>
<div><div>year * required</div><div>integer(\$int32)</div><div>(path)</div></div>	<div>The year of study that the course should be removed from.</div> <div>year</div>
<div><div>courseID * required</div><div>integer(\$int32)</div><div>(path)</div></div>	<div>The course ID of the course that will be removed from the student's schedule.</div> <div>courseID</div>

Responses

Code	Description	Links
200	Success	No links

GET

/api/student/v1/insight/courses/{studentID}

Get reccomeneded courses for student

Parameters

Try it out

Name	Description
<div><div>studentID * required</div><div>integer(\$int32)</div><div>(path)</div></div>	<div>Id of student</div> <div>studentID</div>

schoolId

Id of school of student

integer(\$int32)

(query)

schoolId

Responses

Code	Description	Links
200	Success	No links

University

GET

/api/university/v1/schools

Gets list of universities, with their corresponding ids

Parameters

Try it out

No parameters

Responses

Code	Description	Links
200	Success	No links

Media type

text/plain

Controls Accept header.

Example Value Schema

```
[
  {
    "id": 0,
    "name": "string"
  }
]
```

500	Server Error	No links
-----	--------------	----------

GET

/api/university/v1/programs

Gets list of univesity programs, based on optional filters

^

Parameters

Try it out

Name	Description
UniversityId integer(\$int32) <i>(query)</i>	<div>UniversityId</div>
Search string <i>(query)</i>	<div>Search</div>
count integer(\$int32) <i>(query)</i>	<div>Pagination count</div> <div>count</div>
offset integer(\$int32) <i>(query)</i>	<div>Pagination offset</div> <div>offset</div>

Responses

Code	Description	Links
200	<div>Success</div> <div><div>Media type</div><div>text/plain</div><div>Controls Accept header.</div><div>Example Value Schema</div></div>	No links


```
[
  {
    "id": 0,
    "name": "string",
    "gradeRange": "string",
    "language": "string",
    "prerequisites": "string",
    "website": "string",
    "universityId": 0
  }
]
```

500

Server Error

No links

GET

/api/university/v1/programs/count

Returns the total number of filtered university programs

^

Parameters

Try it out

Name	Description
UniversityId	
integer(\$int32)	
(query)	
Search	
string	
(query)	

Responses

Code	Description	Links
200	Success	No links
	Media type	
	text/plain	
	Controls Accept header.	
	Example Value	Schema
	0	
500	Server Error	No links

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
  name           string
                 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($int32)
          nullable: true
  user      User {...}
  schoolId  integer($int32)
          nullable: true
  school    School {...}
}
```

```
EventCreationDto {
  title      string
            nullable: true
  description string
            nullable: true
  date       string($date-time)
  location   string
            nullable: true
  userId     integer($int32)
}
```

```
Experience {
  isWorkExperience boolean
  startDate integer($int32)
  endDate integer($int32)
  position string
  nullable: true
  company string
  nullable: true
  location string
  nullable: true
  description string
  nullable: true
}

JobApplicationCreation {
  jobPostingId integer($int32)
  userId integer($int32)
  coverLetterId integer($int32)
  nullable: true
}

JobApplicationDetails {
  email string
  nullable: true
  firstName string
  nullable: true
  middleName string
  nullable: true
  lastName string
  nullable: true
  birthday string($date-time)
  nullable: true
  school string
  nullable: true
  grade UserGrade integer($int32)
  Enum:
    Array [ 6 ]
  coverLetter CoverLetter {...}
  skills [...]
  intrestsHobbies [...]
  awards [...]
}

JobPostingCreation {
  userId integer($int32)
  title string
  nullable: true
  description string
  nullable: true
  applicationDeadline string($date-time)
}
```

```
JobPostingDetails {
  postingID      integer($int32)
  firstName      string
                 nullable: true
  middleName     string
                 nullable: true
  lastName       string
                 nullable: true
  companyName    string
                 nullable: true
  country        string
                 nullable: true
  city           string
                 nullable: true
  address        string
                 nullable: true
  companyEmail   string
                 nullable: true
  unitNumber     integer($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 {
  profileId integer($int32)
  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($int32)
              Enum:
                Array [ 3 ]
  affiliation   string
              nullable: true
  affiliationID integer($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

  password       string
                nullable: true
                example: Password1

  Email: A required string representing the user's email address.
  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  string
             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 ]
```