

Student Service API

This is the Student Service API.

Index:

Methods:

Retrieving all students	page2
Retrieving all students in a specific course	page3
Retrieving full info of specific student	page4
Retrieving all courses	page5
Retrieving all courses in a specific year	page6
Retrieving full info of a specific course	page7

Formats:

Available response formats	page8
----------------------------	-------

Metadata:

Status Codes	page9
HTTP Headers	page10
Error Messages	page12

Retrieving all students:

URL: /studentserviceapi

To retrieve a list of all students, you must request by course and use 'all' as the value for the id parameter.

GET Parameter	Value
request	student
by	course
id	all
format	json

Response (Example):

```
{
  "student": [
    {
      "id": "1",
      "name": "Johnny",
      "surname": "Franklin"
    },
    {
      "id": "2",
      "name": "Katie",
      "surname": "Idol"
    },
    {
      "id": "3",
      "name": "Eric",
      "surname": "Idol"
    }
  ]
}
```

Retrieving all students in a specific course:

URL: /studentserviceapi

To retrieve a list of all students for a specific course, you must request by course and use the "course id" as the value for the id parameter. How to get available course ids, luse method 'Retrieve all courses' (see page 5).

GET Parameter	Value
request	student
by	course
id	3 (example)
format	json

Response (Example):

```
{
  "student": [
    {
      "id": "1",
      "name": "Johnny",
      "surname": "Franklin"
    },
    {
      "id": "2",
      "name": "Katie",
      "surname": "Idol"
    },
    {
      "id": "3",
      "name": "Eric",
      "surname": "Idol"
    }
  ]
}
```

Retrieving full info of specific student:

URL: /studentserviceapi

To retrieve the full information of a specific student you must request by student and use the studentId as value for the id parameter. Use method 'Retrieve all students' to find available ids (see page 2).

GET Parameter	Value
request	student
id	3 (example)
format	json

Response (Example):

```
{
  "student": [
    {
      "id": "1",
      "name": "Johnny",
      "surname": "Franklin",
      "age": "37",
      "postAddress": "Bärslanda",
      "streetAddress": "Sexpacket",
      "phonenumbers": [
        {
          "mobile": "0795-561886",
          "home": "016-536482"
        }
      ],
      "course": [
        {
          "id": "3",
          "name": "DB-101_2015",
          "status": "complete",
          "grade": "g"
        },
        {
          "id": "6",
          "name": "C-102_2015",
          "status": "aborted",
          "grade": ""
        }
      ]
    }
  ]
}
```

Retrieving all courses:

URL: /studentserviceapi

To retrieve all courses you must request course by year and use 'all' as value for the year parameter.

GET Parameter	Value
request	course
by	year
id	all
format	json

Response (Example):

```
{
  "course": [
    {
      "id": "1",
      "name": "C-101_2014"
    },
    {
      "id": "2",
      "name": "JAVA-101_2015"
    },
    {
      "id": "3",
      "name": "DB-101_2015"
    }
  ]
}
```

Retrieving all courses in a specific year:

URL: /studentserviceapi

To retrieve all courses in a specific year you need to request course by year and use the year (YYYY) as value for the id parameter.

GET Parameter	Value
request	course
by	year
id	2016 (example)
format	json

Response (Example)

```
{
  "course": [
    {
      "id": "1",
      "name": "C-101_2014"
    },
    {
      "id": "2",
      "name": "JAVA-101_2015"
    },
    {
      "id": "3",
      "name": "DB-101_2015"
    }
  ]
}
```

Retrieving full info of a specific course:

URL: /studentserviceapi

To retrieve detailed information about a specific course request course and use courseId as value for the id parameter. Use method 'Retrieve all courses' (see page 5) to find available courseId:s.

GET Parameter	Value
request	course
id	3 (example)
format	json

Response (Example):

```
{
  "course": [
    {
      "id": "1",
      "name": "C-101_2014",
      "description": "Programming with C",
      "startDate": "2014-02-22",
      "endDate": "2014-11-14",
      "points": "60",
      "student": [
        {
          "id": "50",
          "name": "Eva",
          "surname": "Croce",
          "status": "complete",
          "grade": "ig"
        },
        {
          "id": "76",
          "name": "Tom",
          "surname": "Lennon",
          "status": "complete",
          "grade": "g"
        }
      ]
    }
  ]
}
```

Available response formats

JSON

Example data:

```
{
  "student": [
    {
      "id": "8",
      "name": "Jim",
      "surname": "James"
    },
    {
      "id": "97",
      "name": "Roky",
      "surname": "Graves"
    }
  ]
}
```

XML

Example data:

```
<ITEMS>
  <students>
    <student id="8">
      <id>8</id>
      <name>Jim</name>
      <surname>James</surname>
    </student>
    <student id="97">
      <id>97</id>
      <name>Roky</name>
      <surname>Graves</surname>
    </student>
  </students>
</ITEMS>
```

HTML

Example data:

```
<html>
  <head>
    <title>Requested data</title>
  </head>
  <body>
    <p>student</p>
    <table border="1">
      <th>id</th><th>name</th><th>surname</th>
      <tr><td>8</td><td>Jim</td><td>James</td></tr>
      <tr><td>97</td><td>Roky</td><td>Graves</td></tr>
    </table>
```


Status codes:

200 = OK.

400 = Bad request, check get parameters.

404 = No data matching query.

415 = Requested Format Not Supported.

500 = Internal Server Error.

HTTP Headers:

OK

200 OK
Connection: Keep-Alive
Date: Thu, 27 Apr 2017 12:32:30 GMT
Server: Winstone Servlet Engine v0.9.10
Content-Length: 1648
Content-Type: application/xml;charset=UTF-8
Client-Date: Thu, 27 Apr 2017 12:32:30 GMT
Client-Peer: 127.0.0.1:8080
Client-Response-Num: 1
X-Powered-By: Servlet/2.5 (Winstone/0.9.10)

Bad request

400 Bad Request
Connection: Keep-Alive
Date: Thu, 27 Apr 2017 12:38:49 GMT
Server: Winstone Servlet Engine v0.9.10
Content-Length: 110
Content-Type: application/xml;charset=UTF-8
Client-Date: Thu, 27 Apr 2017 12:38:49 GMT
Client-Peer: 127.0.0.1:8080
Client-Response-Num: 1
X-Powered-By: Servlet/2.5 (Winstone/0.9.10)

No data matching query

404 Not Found
Connection: Keep-Alive
Date: Thu, 27 Apr 2017 12:39:43 GMT
Server: Winstone Servlet Engine v0.9.10
Content-Length: 113
Content-Type: application/xml;charset=UTF-8
Client-Date: Thu, 27 Apr 2017 12:39:43 GMT
Client-Peer: 127.0.0.1:8080
Client-Response-Num: 1
X-Powered-By: Servlet/2.5 (Winstone/0.9.10)

Requested Format Not Supported

415 Unsupported Media Type
Connection: Keep-Alive
Date: Wed, 10 May 2017 13:34:44 GMT
Server: Winstone Servlet Engine v0.9.10
Content-Length: 33
Content-Type: text/plain;charset=ISO-8859-1
Client-Date: Wed, 10 May 2017 13:34:44 GMT
Client-Peer: 127.0.0.1:8080
Client-Response-Num: 1
X-Powered-By: Servlet/2.5 (Winstone/0.9.10)

Internal Server Error

500 Internal Server Error
Connection: Keep-Alive
Date: Wed, 10 May 2017 14:29:05 GMT
Server: Winstone Servlet Engine v0.9.10
Content-Length: 129
Content-Type: application/xml;charset=UTF-8
Client-Date: Wed, 10 May 2017 14:29:05 GMT
Client-Peer: 127.0.0.1:8080
Client-Response-Num: 1
X-Powered-By: Servlet/2.5 (Winstone/0.9.10)

Error Messages:

Here are some examples of error responses from the API. Error responses will be formatted to requested type if possible, otherwise text/plain will be returned.

400 Bad Request:

- **xml:**

```
<error>
  <message>Bad Request. Invalid parameters</message>
</error>
```

- **json:**

```
{"error":[{"message":"Bad Request. Invalid parameters"}]}
```

- **html:**

```
<html>
  <head>
    <title>Requested data</title>
  </head>
  <body>
    <p>Bad Request. Invalid parameters</p>
  </body>
</html>
```

404 Not Found:

- **xml:**

```
<error>
  <message>No data available for requested id</message>
</error>
```

- **json:**

```
{"error":[{"message":"No data available for requested id"}]}
```

- **html:**

```
<html>
  <head>
    <title>Requested data</title>
  </head>
  <body>
    <p>No data available for requested id</p>
  </body>
</html>
```

415 Unsupported Media Type:

- **text/plain:**

```
Requested format not supported.
```

500 Internal Server Error

- **xml:**

```
<error>
  <message>Internal server error: Could not connect to storage</message>
</error>
```

- **json:**

```
{"error":[{"message":"Internal server error: Could not connect to
storage"}]}
```

- **html:**

```
<html>
  <head>
    <title>Requested data</title>
  </head>
  <body>
    <p>Internal server error: Could not connect to storage</p>
  </body>
</html>
```