



# Course REST-API <sup>1.0.0</sup>

[ Base URL: `courses.de/` ]

A presentation about a theoretical courses REST-API.

Worktime till now: 3h

[Contact the developer](#)

## Course



**GET**    **/courses**    Get all courses



### Parameters

Try it out

Name	Description
------	-------------

timeRange	Time frame in which the courses take place. Example: 1621948000-1621950000
-----------	---

timeRange - Time frame in which the courses take place.Exar

### Responses

Response content type

application/json

Code	Description
------	-------------

**Code****Description**

200

Returns all courses

**Example Value**    Model

```
[
  {
    "id": "1",
    "name": "How to create a Restful API",
    "profName": "Linus Torvalds",
    "description": "This time we will discuss how we can plan and create a restful API in go",
    "price": "00.00€",
    "online": true,
    "date": 1621949400
  }
]
```

**POST****/courses**    Add a course**Parameters**[Try it out](#)

No parameters

**Responses**

Response content type

[application/json](#)**Code****Description**

200

Course Added

**Example Value**    Model

```
{
  "id": "1",
  "name": "How to create a Restful API",
  "profName": "Linus Torvalds",
  "description": "This time we will discuss how we can plan and create a restful API in go",
  "price": "00.00€",
  "online": true,
  "date": 1621949400
}
```

**Code****Description**

400

Validation failed

**Example Value**    Model

```
{
  "errorCode": 2231,
  "errorDescription": "Validation failed"
}
```

**GET****/courses/{courseId}** Finds Course by ID**Parameters****Try it out****Name****Description****courseId** \* required

Course ID

integer(\$int32)

(path)

courseId - Course ID

**Responses**

Response content type

**application/json****Code****Description**

200

found Course

**Example Value**    Model

```
{
  "id": "1",
  "name": "How to create a Restful API",
  "profName": "Linus Torvalds",
  "description": "This time we will discuss how we can plan and create a restful API in go",
  "price": "00.00€",
  "online": true,
  "date": 1621949400
}
```

PUT

/courses/{courseId} Update a course



## Parameters

Try it out

Name

Description

**courseId** \* required

integer(\$int32)

(path)

courseId

## Responses

Response content type

application/json

Code

Description

200

course updated

Example Value Model

```
{
  "id": "1",
  "name": "How to create a Restful API",
  "profName": "Linus Torvalds",
  "description": "This time we will discuss how we can plan and create a rest
ful API in go",
  "price": "00.00€",
  "online": true,
  "date": 1621949400
}
```

400

Course does not exist

Example Value Model

```
{
  "errorCode": 1337,
  "errorDescription": "Entity does not exist"
}
```

POST

/courses/{courseId}/participate Add yourself as a participant



## Parameters

Try it out

Name	Description
------	-------------

<b>courseId</b> * required	
----------------------------	--

integer(\$int32)	
------------------	--

(path)	
--------	--

courseId
----------

<b>userId</b> * required	
--------------------------	--

integer(\$int32)	
------------------	--

(header)	
----------	--

userId
--------

Responses	
-----------	--

Response content type	
-----------------------	--

application/json
------------------

Code	Description
------	-------------

204	
-----	--

User Added
------------

400	
-----	--

Seats full / course does not exist
------------------------------------

Example Value	Model
---------------	-------

<pre>{   "errorCode": 3321,   "errorDescription": "All seats are taken" }</pre>	
---	--

Models
--------





```
AllCourses  [AllCourses  {
    id                string
                     example: 1
    name              string
                     example: How to create a Restful API

                     Name of the course

    profName          string
                     example: Linus Torvalds

                     Name of the professor

    description        string
                     example: This time we will discuss how we can plan and create
                     a restful API in go

                     Short description of the course

    price              string
                     example: 00.00€

                     Price of the course. Must follow this pattern: 100.00€

    online             boolean
                     example: true

                     If the course is online or not

    date              integer($int32)
                     example: 1621949400
}]
```

Course {

id string  
example: 1

name string  
example: How to create a Restful API

Name of the course

profName string  
example: Linus Torvalds

Name of the professor

description string  
example: This time we will discuss how we can plan and create a restful API in go

Short description of the course

price string  
example: 00.00€

Price of the course. Must follow this pattern: 100.00€

online boolean  
example: true

If the course is online or not

date integer(\$int32)  
example: 1621949400

}

EntityDoesNotExistError {

errorCode integer(\$int32)  
example: 1337

errorDescription string  
example: Entity does not exist

}

SeatsFullError {

errorCode integer(\$int32)  
example: 3321

errorDescription string  
example: All seats are taken

}



```
ValidationFailedError {  
  errorCode      integer($int32)  
                  example: 2231  
  errorDescription string  
                  example: Validation failed  
}
```