Dokumentation Erwachsenenbildung Rest-API

Einführung 3
Aufruf der Rest-API 3
Response Codes 3
Course 4
/rest/course/{uid} 4
/rest/course/category/{uid} 5
/rest/course/educationalInstruction/{uid} 6
/rest/course/educationalInstruction/{uid}/complete 6
/rest/course/search/{searchString} 7
/rest/course/category/{uid}/search/{searchString} 7
/rest/course/{apiKey}/create 8
/rest/course/{apiKey}/create 11
/rest/course/{apiKey}/create/uid 12
/rest/course/{apiKey}/create/uid 15
/rest/course/{apiKey}/update 16
/rest/course/{apiKey}/update 19
/rest/course/{apiKey}/update/uid 20
/rest/course/{apiKey}/update/uid 23
/rest/course/filter 24
/rest/course/filter 26
/rest/course/{apiKey}/{courseNumber} 26
/rest/course/{apiKey}/{courseNumber} 26
/rest/course/{uid} 26
/rest/course/{uid} 26
EducationalInstruction 27
/rest/educationalInstruction 27
/rest/educationalInstruction/{apiKey} 27
Category 28
/rest/category 28
Meeting 29
/rest/meeting 29
/rest/meeting/educationalInstruction/{uid} 29

Instructor 30
/rest/instructor/educationalInstruction/{uid} 30
/rest/instructor/{apiKey}/create 30
/rest/instructor/{apiKey}/create 31
Mail 32
/rest/mail/key 32
/rest/mail/send 32

Einführung

Allgemein | Aufruf der Rest-API

Die Rest-API stellt verschiedene Controller zur Verfügung, die wiederum Methoden nach außen zugänglich machen. Eine solche Methode wird aufgerufen, indem die dafür eindeutig definierte URL angesteuert wird. Diese kann am Beispiel der Abfrage eines Kurses wie folgt aussehen:

http://beispieladresse:beispielport/rest/course/1

Wichtig hierbei ist, dass der Anfragetyp (GET, POST, DELETE) zur Methode passt. Im Beispiel müsste die Anfrage demnach als GET-Anfrage gesendet werden. Innerhalb der Anfragetypen gibt es folgende Unterscheidungen:

- GET-Anfragen werden gestellt, um Daten abzufragen. Im Normalfall werden hierbei benötigte Daten als Parameter in der URL übergeben.
- POST-Anfragen werden gestellt, um Daten zu übermitteln und entsprechend neue Datensätze anzulegen oder bestehende Datensätze zu aktualisieren. Hierbei können Daten ebenfalls als Parameter in der URL übermittelt werden. Allerdings ist die bevorzugte Variante die Übermittlung der Daten im Body der Anfrage. Dabei werden die Daten im ***JSON-Format*** übergeben.
- DELETE-Anfragen werden gestellt, um Daten zu löschen. Wie bei den POST-Anfragen ist sowohl das Übermitteln der Daten als Parameter in der URL als auch das Senden der Daten im Body möglich.

Wie die jeweiligen Parameter der zur Verfügung gestellten Methoden im Detail aussehen, kann im weiteren Verlauf der Dokumentation eingesehen werden.

Allgemein | Response Codes

Die verschiedenen Aufrufe der Rest-API liefern verschiedene Antwortcodes, die davon abhängen, ob bei der Bearbeitung Fehler aufgetreten sind und um welche Fehler es sich handelt. Innerhalb der Anwendung können folgende Response Codes auftreten:

- **200**: Dieser Code bedeutete, dass kein Fehler aufgetreten ist und ein Ergebnis geliefert wurde.
- **404**: Dieser Code wird innerhalb der Anwendung an verschiedenen Stellen genutzt. Er tritt auf, wenn eine Anfrage nicht korrekt ist oder auch, wenn zur Anfrage keine Ressourcen gefunden werden. In jedem Fall wird in der Fehlernachricht der Fehler ausgegeben.
- **501**: Wenn dieser Fehler auftritt, ist bei der Verarbeitung des Aufrufs ein Fehler aufgetreten. Dies kann verschiedene Gründe haben, wie einen falschen Datentyp im Parameter.

Welche Response Codes in welchen Anfragen auftreten, ist für jede Anfrage nochmals aufgeschlüsselt.

Course

GET	/rest/course/{uid}

Gibt den Kurs zur angegebenen eindeutigen ID im Parameter {uid} zurück.

Parameter:

uid: „int“	Eindeutige ID des gesuchten Kurses

Beispiel:

uid = 1

Antwort:

Response Code 200: [{

„courseNumber“: „string“,	Kursnummer
„titel“: „string“,	Titel
„description“: „string“,	Beschreibung
„uid“: „int“,	Eindeutige ID des Kurses
„startTime“: „datetime“,	Beginn
„endTime“: „datetime“,	Ende
„externalLink“: „string“,	Externer Link
„learningLink“: „string“,	Learning Link
„price“: „decimal“,	Preis
„examPrice“: „decimal“,	Prüfungspreis
„participantCount“: „int“,	Teilnehmerzahl
„concept“: „string“,	Konzept
„instructors“: [{	Liste der Dozenten
„instructorNumber“: „string“,	Dozentennummer
„gender“: „string“,	Anrede
„firstName“: „string“,	Vorname
„lastName“: „string“,	Nachname
„email“: „string“,	E-Mail
„job“: „string“,	Beruf
„mobil“: „string“,	Handynummer
„tel“: „string“,	Telefonnummer
„fax“: „string“,	Fax
„uid“: „int“	Eindeutige ID des Dozenten
}],	
„educationalInstruction“: {	Bildungseinrichtung
„name“: „string“,	Name
„street“: „string“,	Straße
„city“: „string“,	Stadt
„zipCode“: „string“,	Postleitzahl
„fallbackLink“: „string“,	Fallback Link
„tel“: „string“,	Telefonnummer
„logo“: „string“	Logo
},	

„categorys“: [{	Liste der Kategorien
„titel“: „string“,	Kategorietitel
„description“: „string“,	Kategoriebeschreibung
„uid“: „int“	Eindeutige ID der Kategorie
}],	
„meetings“: [{	Liste der Termine
“startTime“: „datetime“,	Beginn
„endTime“: „datetime“,	Ende
„hours“: „int“,	Anzahl der Stunden
„instructors“: [{	Liste der Dozenten des Termins
„instructorNumber“: „string“,	Dozentennummer
„gender“: „string“,	Anrede
„firstName“: „string“,	Vorname
„lastName“: „string“,	Nachname
„email“: „string“,	E-Mail
„job“: „string“,	Beruf
„mobil“: „string“,	Handynummer
„tel“: „string“,	Telefonnummer
„fax“: „string“,	Fax
„uid“: „int“	Eindeutige ID des Dozenten
}],	
„uid“: „int“	Eindeutige ID des Termins
}],	
„buildingName“: „string“,	Gebäudename
„street“: „string“,	Straße
„zipCode“: „string“,	Stadt
„city“: „string“,	Postleitzahl
„lat“: „decimal“,	Lat
„lng“: „decimal“,	Lng
„totalHours“: „int“	Gesamtzahl Stunden
}]	

Response Code 404: Es wurde kein Kurs mit der angegebenen ID gefunden

GET	/rest/course/category/{uid}

Gibt alle Kurse zur Kategorie mit der angegebenen ID im Parameter {uid} zurück.

Parameter:

uid: „int“	Eindeutige ID der zu durchsuchenden Kategorie

Beispiel:

uid = 1

Antwort:

Response Code 200: [{

„courseNumber“: „string“,	Kursnummer
„titel“: „string“,	Titel
„description“: „string“,	Beschreibung
„uid“: „int“	Eindeutige ID des Kurses

}]

GET	/rest/course/educationalInstruction/{uid}
Gibt alle Kurse zur Bildungseinrichtung mit der angegebenen ID im Parameter {uid} abgespeckt zurück.	
Parameter:	
uid: „int“	Eindeutige ID der Bildungseinrichtung
Beispiel:	
uid = 1	
Antwort:	
Response Code 200: [{	Rückgabe, wenn ein Kurs gefunden wurde
„courseNumber“: „string“,	Kursnummer
„titel“: „string“,	Titel
„description“: „string“,	Beschreibung
„uid“: „int“	Eindeutige ID des Kurses
}]	
Response Code 404: Es wurde keine Bildungseinrichtung zur angegebenen ID gefunden	

GET	/rest/course/educationalInstruction/{uid}/complete
Gibt alle Kurse zur Bildungseinrichtung mit der angegebenen ID im Parameter {uid} vollständig zurück.	
Parameter:	
uid: „int“	Eindeutige ID der Bildungseinrichtung
Beispiel:	
uid = 1	
Antwort:	
Response Code 200: [{	Rückgabe, wenn ein Kurs gefunden wurde
„courseNumber“: „string“,	Kursnummer
„titel“: „string“,	Titel
„description“: „string“,	Beschreibung
„uid“: „int“	Eindeutige ID des Kurses
„startTime“: „datetime“,	Beginn
„endTime“: „datetime“,	Ende
„externalLink“: „string“,	Externer Link
„learningLink“: „string“,	Learning Link
„price“: „decimal“,	Preis
„examPrice“: „decimal“,	Prüfungspreis
„participantCount“: „int“,	Teilnehmerzahl
„concept“: „string“,	Konzept
„buildingName“: „string“,	Gebäudename
„street“: „string“,	Straße
„zipCode“: „string“,	Stadt
„city“: „string“,	Postleitzahl
„lat“: „string“,	Lat
„lng“: „string“,	Lng
„totalHours“: „int“	Gesamtzahl Stunden
}]	
Response Code 404: Es wurde keine Bildungseinrichtung zur angegebenen ID gefunden	

GET	/rest/course/search/{searchString}

Sucht alle Kurse, die dem Suchstring entsprechen. Hierbei gibt es mehrere Möglichkeiten, wann ein Kurs zurückgeliefert wird. Die erste Möglichkeit ist, dass der Suchstring der Kursnummer entspricht. Alternativ kann der Suchstring auch im Titel oder der Beschreibung enthalten sein.

Parameter:

searchString: „string“	Suchanfrage

Beispiel:

searchString = „Mustersuche“

Antwort:

Response Code 200: [{

„courseNumber“: „string“,	Kursnummer
„titel“: „string“,	Titel
„description“: „string“,	Beschreibung
„uid“: „int“	Eindeutige ID des Kurses

}]

GET	/rest/course/category/{uid}/search/{searchString}

Sucht alle Kurse innerhalb der Kategorie mit der angegebenen ID im Parameter {uid}, die dem Suchstring entsprechen. Hierbei gibt es mehrere Möglichkeiten, wann ein Kurs zurückgeliefert wird. Die erste Möglichkeit ist, dass der Suchstring der Kursnummer entspricht. Alternativ kann der Suchstring auch im Titel oder der Beschreibung enthalten sein.

Parameter:

uid: „int“,	Eindeutige ID der zu durchsuchenden Kategorie
searchString: „string“	Suchanfrage

Beispiel:

uid = 1,
searchString = „Mustersuche“

Antwort:

Response Code 200: [{

„courseNumber“: „string“,	Kursnummer
„titel“: „string“,	Titel
„description“: „string“,	Beschreibung
„uid“: „int“	Eindeutige ID des Kurses

}]

POST	/rest/course/{apiKey}/create

Erstellt einen neuen Kurs für die im Parameter {apiKey} spezifizierte Bildungseinrichtung. Vor der Verwendung der Methode, muss es einen Applikationsschlüssel für die jeweilige Bildungseinrichtung geben. Die Dozenten können auf verschiedene Arten angegeben werden, was sowohl für die Dozenten des Kurses als auch der jeweiligen Termine gilt. Zuerst ist eine Liste von Nummern angebbar, wodurch die Dozenten anhand der Dozentnummern gesucht und zugeordnet werden. Weiterhin ist es möglich, Dozentenobjekte zu übergeben. In diesem Fall wird zuerst geschaut, ob es die Dozentennummer bereits gibt. Falls nicht, wird der Dozent angelegt und dann zugeordnet.
Auch die Datumswerte können auf verschiedene Arten angegeben werden. Möglich sind die Schreibweise nach ISO sowie nach DIN (Beispiele: 1970-01-01T00:00:00+00:00, 01.01.1970 00:00:00+00:00, …).
Die Kategorien werden anhand von Schlagworten ausgewertet. Kommt ein Schlagwort in einer Kategorie vor, wird der Kurs dieser zugeordnet. Alternativ können hier auch komplette Kategorieobjekte übergeben werden. In diesem Fall wird der Titel quasi als Schlagwort verwendet und ist ausschlaggebend für die Zuordnung der Kategorie zum Kurs.
Pflichtfelder beim Anlegen sind die Kursnummer, der Titel sowie Beginn und Ende des Kurses.

Parameter:

```
{
    „courseNumber“: „string“,                Kursnummer
    „titel“: „string“,                       Titel
    „description“: „string“,                 Beschreibung
    „startTime“: „datetime“,                 Beginn
    „endTime“: „datetime“,                   Ende
    „externalLink“: „string“,                Externer Link
    „learningLink“: „string“,                Learning Link
    „price“: „decimal“,                      Preis
    „examPrice“: „decimal“,                  Prüfungspreis
    „participantCount“: „int“,               Teilnehmerzahl
    „concept“: „string“,                     Konzept
    „instructors“: [„int“],                  Liste der Dozenten
    „categorys“: [„string“],                 Liste der Kategorien
    „meetings“: [{                           Liste der Termine
        „startTime“: „datetime“,                 Beginn
        „endTime“: „datetime“,                   Ende
        „hours“: „int“,                          Anzahl der Stunden
        „instructors“: [„int“]                   Liste der Dozenten
    }],
    „buildingName“: „string“,                Gebäudename
    „street“: „string“,                      Straße
    „city“: „string“,                        Stadt
    „zipCode“: „string“,                     Postleitzahl
    „totalHours“: „int“                      Total Hours
}
```

Beispiel: Übergabe von IDs

```
{
    „courseNumber“: „181-4B04“,
    „titel“: „Musterkurs“,
    „description“: „Musterbeschreibung“,
    „startTime“: „2020-02-28 08:00“,
    „endTime“: „2020-02-28 18:00“,
    „externalLink“: „Musterlink“,
    „learningLink“: „Musterlink“,
    „price“: 50.5,
    „examPrice“: 100,
    „participantCount“: 20,
    „concept“: „Musterkonzept“,
    „instructors“: [67, 204],
    „categorys“: [„Muster1“, „Muster2],
    „meetings“: [{
        „startTime“: „2020-02-28 08:00“,
        „endTime“: „2020-02-28 18:00“,
        „instructors“: [67]
    }],
    „buildingName“: „Mustergebäude“,
    „street“: „Musterstraße“,
    „city“: „Musterstadt“,
    „zipCode“: „Musterpostleitzahl“,
    „totalHours“: 10
}
```

Beispiel: Übergabe von Objekten

```
{
    „courseNumber“: „181-4B04“,
    „titel“: „Musterkurs“,
    „description“: „Musterbeschreibung“,
    „startTime“: „2020-02-28 08:00“,
    „endTime“: „2020-02-28 18:00“,
    „externalLink“: „Musterlink“,
    „learningLink“: „Musterlink“,
    „price“: 50.5,
    „examPrice“: 100,
    „participantCount“: 20,
    „concept“: „Musterkonzept“,
    „instructors“: [{
        „instructorNumber“: „67“,
        „gender“: „M“,
        „firstName“: „Max“,
        „lastName“: „Mustermann“,
        „email“: „Mustermail“,
        „job“: „Musterberuf“,
        „mobil“: „12345“,
        „tel“: „12345“,
        „fax“: „12345“
    }],
    „categorys“: [{
        „titel“: „Muster“,
        „description“: „Musterbeschreibung“
    }],
    „meetings“: [{
        „startTime“: „2020-02-28 08:00“,
        „endTime“: „2020-02-28 18:00“,
        „hours“: 10,
        „instructors“: [67]
    }],
    „buildingName“: „Mustergebäude“,
    „street“: „Musterstraße“,
    „city“: „Musterstadt“,
    „zipCode“: „Musterpostleitzahl“,
    „totalHours“: 10
}
```

Antwort:

Response Code 200: {

„courseNumber“: „string“,	Kursnummer
„titel“: „string“,	Titel
„description“: „string“,	Beschreibung
„uid“: „int“,	Eindeutige ID des Kurses
„startTime“: „datetime“,	Beginn
„endTime“: „datetime“,	Ende
„externalLink“: „string“,	Externer Link
„learningLink“: „string“,	Learning Link
„price“: „decimal“,	Preis
„examPrice“: „decimal“,	Prüfungspreis
„participantCount“: „int“,	Teilnehmerzahl
„concept“: „string“,	Konzept
„instructors“: [{	Liste der Dozenten
„instructorNumber“: „string“,	Dozentennummer
„gender“: „string“,	Anrede
„firstName“: „string“,	Vorname
„lastName“: „string“,	Nachname
„email“: „string“,	E-Mail
„job“: „string“,	Beruf
„mobil“: „string“,	Handynummer
„tel“: „string“,	Telefonnummer
„fax“: „string“,	Fax
„uid“: „int“	Eindeutige ID des Dozenten
}],	
„educationalInstruction“: {	Bildungseinrichtung
„name“: „string“,	Name
„street“: „string“,	Straße
„city“: „string“,	Stadt
„zipCode“: „string“,	Postleitzahl
„fallbackLink“: „string“,	Fallback Link
„tel“: „string“,	Telefonnummer
„logo“: „string“	Logo
},	
„categorys“: [{	Liste der Kategorien
„titel“: „string“,	Kategorietitel
„description“: „string“,	Kategoriebeschreibung
„uid“: „int“	Eindeutige ID der Kategorie
}],	

„meetings“: [{	Liste der Termine
“startTime“: „datetime“,	Beginn
„endTime“: „datetime“,	Ende
„hours“: „int“,	Anzahl der Stunden
„instructors“: [{	Liste der Dozenten des Termins
„instructorNumber“: „string“,	Dozentennummer
„gender“: „string“,	Anrede
„firstName“: „string“,	Vorname
„lastName“: „string“,	Nachname
„email“: „string“,	E-Mail
„job“: „string“,	Beruf
„mobil“: „string“,	Handynummer
„tel“: „string“,	Telefonnummer
„fax“: „string“,	Fax
„uid“: „int“	Eindeutige ID des Dozenten
}],	
„uid“: „int“	Eindeutige ID des Termins
}],	
„buildingName“: „string“,	Gebäudename
„street“: „string“,	Straße
„zipCode“: „string“,	Stadt
„city“: „string“,	Postleitzahl
„lat“: „decimal“,	Lat
„lng“: „decimal“,	Lng
„totalHours“: „int“	Gesamtzahl Stunden
}	

Response Code 404: Es existiert bereits ein Kurs / Keine Bildungseinrichtung für den ApiKey

OPTIONS	/rest/course/{apiKey}/create

Diese Abfrage soll nicht direkt von außen angesprochen werden. Die Abfrage ist dafür da, für die zugehörige POST-Methode die CORS-Richtlinien zu prüfen und wird dafür von modernen Browsern vor der POST-Abfrage aufgerufen.

POST	/rest/course/{apiKey}/create/uid

Erstellt einen neuen Kurs für die im Parameter {apiKey} spezifizierte Bildungseinrichtung. Vor der Verwendung der Methode, muss es einen Applikationsschlüssel für die jeweilige Bildungseinrichtung geben. Dozenten und Termine können nur als Objekte angegeben werden. Ist eine *uid* enthalten, wird der das Objekt angepasst, sonst wird es erstellt und zugeordnet.
Datumswerte können auf verschiedene Arten angegeben werden. Möglich sind die Schreibweise nach ISO oder DIN (Beispiele: 1970-01-01T00:00:00+00:00, 01.01.1970 00:00:00+00:00, …).
Die Kategorien werden anhand von Schlagworten ausgewertet. Kommt ein Schlagwort in einer Kategorie vor, wird der Kurs zugeordnet. Alternativ können vollständige Kategorieobjekte übergeben werden. Pflichtfelder beim Anlegen sind die Kursnummer, der Titel sowie Beginn und Ende des Kurses.

Parameter:

```
{
    „courseNumber“: „string“,                  Kursnummer
    „titel“: „string“,                         Titel
    „description“: „string“,                   Beschreibung
    „startTime“: „datetime“,                   Beginn
    „endTime“: „datetime“,                     Ende
    „externalLink“: „string“,                  Externer Link
    „learningLink“: „string“,                  Learning Link
    „price“: „decimal“,                        Preis
    „examPrice“: „decimal“,                    Prüfungspreis
    „participantCount: „int“,                  Teilnehmerzahl
    „concept“: „string“,                       Konzept
    „instructors“: [{                          Liste der Dozenten
        „instructorNumber“: „string“,              Dozentennummer
        „gender“: „string“,                        Anrede/Geschlecht
        „firstName“: „string“,                     Vorname
        „lastName“: „string“,                      Nachname
        „email“: „string“,                         E-Mail
        „job“: „string“,                           Beruf
        „mobil“: „string“,                         Mobilnummer
        „tel“: „string“,                           Telefonnummer
        „fax“: „string“,                           Fax
        „uid“: „int“                               Uid
    }],
    „categorys“: [„string“],                   Liste der Kategorien
    „meetings“: [{                             Liste der Termine
        „startTime“: „datetime“,                   Beginn
        „endTime“: „datetime“,                     Ende
        „hours“: „int“,                            Anzahl der Stunden
        „instructors“: [„instructor“]              Liste der Dozenten
    }],
    „buildingName“: „string“,                  Gebäudename
    „street“: „string“,                        Straße
    „city“: „string“,                          Stadt
    „zipCode“: „string“,                       Postleitzahl
    „totalHours“: „int“                        Total Hours
}
```

Beispiel:


```
{
    „courseNumber“: „181-4B04“,
    „titel“: „Musterkurs“,
    „description“: „Musterbeschreibung“,
    „startTime“: „2020-02-28 08:00“,
    „endTime“: „2020-02-28 18:00“,
    „externalLink“: „Musterlink“,
    „learningLink“: „Musterlink“,
    „price“: 50.5,
    „examPrice“: 100,
    „participantCount: 20,
    „concept“: „Musterkonzept“,
    „instructors“: [{
        „instructorNumber“: „67“,
        „gender“: „M“,
        „firstName“: „Max“,
        „lastName“: „Mustermann“,
        „email“: „Mustermail“,
        „job“: „Musterberuf“,
        „mobil“: „12345“,
        „tel“: „12345“,
        „fax“: „12345“,
        „uid“: 123
    }],
    „categorys“: [„Muster1“, „Muster2“],
    „meetings“: [{
        „startTime“: „2020-02-28 08:00“,
        „endTime“: „2020-02-28 18:00“,
        „hours“:
        „instructors“: [[{
            „instructorNumber“: „67“,
            „gender“: „M“,
            „firstName“: „Max“,
            „lastName“: „Mustermann“,
            „email“: „Mustermail“,
            „job“: „Musterberuf“,
            „mobil“: „12345“,
            „tel“: „12345“,
            „fax“: „12345“,
            „uid“: 123
        }]
    }],
    „buildingName“: „Mustergebäude“,
    „street“: „Musterstraße“,
    „city“: „Musterstadt“,
    „zipCode“: „Musterpostleitzahl“,
    „totalHours“: 10
}
```

Antwort:

Response Code 200: {

„courseNumber“: „string“,	Kursnummer
„titel“: „string“,	Titel
„description“: „string“,	Beschreibung
„uid“: „int“,	Eindeutige ID des Kurses
„startTime“: „datetime“,	Beginn
„endTime“: „datetime“,	Ende
„externalLink“: „string“,	Externer Link
„learningLink“: „string“,	Learning Link
„price“: „decimal“,	Preis
„examPrice“: „decimal“,	Prüfungspreis
„participantCount“: „int“,	Teilnehmerzahl
„concept“: „string“,	Konzept
„instructors“: [{	Liste der Dozenten
„instructorNumber“: „string“,	Dozentennummer
„gender“: „string“,	Anrede
„firstName“: „string“,	Vorname
„lastName“: „string“,	Nachname
„email“: „string“,	E-Mail
„job“: „string“,	Beruf
„mobil“: „string“,	Handynummer
„tel“: „string“,	Telefonnummer
„fax“: „string“,	Fax
„uid“: „int“	Eindeutige ID des Dozenten
}],	
„educationalInstruction“: {	Bildungseinrichtung
„name“: „string“,	Name
„street“: „string“,	Straße
„city“: „string“,	Stadt
„zipCode“: „string“,	Postleitzahl
„fallbackLink“: „string“,	Fallback Link
„tel“: „string“,	Telefonnummer
„logo“: „string“	Logo
},	
„categorys“: [{	Liste der Kategorien
„titel“: „string“,	Kategorietitel
„description“: „string“,	Kategoriebeschreibung
„uid“: „int“	Eindeutige ID der Kategorie
}],	

„meetings“: [{	Liste der Termine
“startTime“: „datetime“,	Beginn
„endTime“: „datetime“,	Ende
„hours“: „int“,	Anzahl der Stunden
„instructors“: [{	Liste der Dozenten des Termins
„instructorNumber“: „string“,	Dozentennummer
„gender“: „string“,	Anrede
„firstName“: „string“,	Vorname
„lastName“: „string“,	Nachname
„email“: „string“,	E-Mail
„job“: „string“,	Beruf
„mobil“: „string“,	Handynummer
„tel“: „string“,	Telefonnummer
„fax“: „string“,	Fax
„uid“: „int“	Eindeutige ID des Dozenten
}],	
„uid“: „int“	Eindeutige ID des Termins
}],	
„buildingName“: „string“,	Gebäudename
„street“: „string“,	Straße
„zipCode“: „string“,	Stadt
„city“: „string“,	Postleitzahl
„lat“: „decimal“,	Lat
„lng“: „decimal“,	Lng
„totalHours“: „int“	Gesamtzahl Stunden
}	

Response Code 404: Es existiert bereits ein Kurs / Keine Bildungseinrichtung für den ApiKey

OPTIONS	/rest/course/{apiKey}/create/uid

Diese Abfrage soll nicht direkt von außen angesprochen werden. Die Abfrage ist dafür da, für die zugehörige POST-Methode die CORS-Richtlinien zu prüfen und wird dafür von modernen Browsern vor der POST-Abfrage aufgerufen.

POST	/rest/course/{apiKey}/update

Aktualisiert einen Kurs für die im Parameter {apiKey} spezifizierte Bildungseinrichtung anhand der Kursnummer. Vor der Verwendung der Methode, muss es einen Applikationsschlüssel für die jeweilige Bildungseinrichtung geben. Die Dozenten können auf verschiedene Arten angegeben werden, was sowohl für die Dozenten des Kurses als auch der jeweiligen Termine gilt. Zuerst ist eine Liste von Nummern angebbar, wodurch die Dozenten anhand der Dozentnummern gesucht und zugeordnet werden. Weiterhin ist es möglich, Dozentenobjekte zu übergeben. In diesem Fall wird zuerst geschaut, ob es die Dozentennummer bereits gibt. Falls nicht, wird der Dozent angelegt und dann zugeordnet.
Auch die Datumswerte können auf verschiedene Arten angegeben werden. Möglich sind die Schreibweise nach ISO oder DIN (Beispiele: 1970-01-01T00:00:00+00:00, 01.01.1970 00:00:00+00:00, …).
Die Kategorien werden anhand von Schlagworten ausgewertet. Kommt ein Schlagwort in einer Kategorie vor, wird der Kurs zugeordnet. Alternativ können vollständige Kategorieobjekte übergeben werden.
Pflichtfelder beim Anlegen sind die Kursnummer, der Titel sowie Beginn und Ende des Kurses.

Parameter:

```
{
    „courseNumber“: „string“,                Kursnummer
    „titel“: „string“,                       Titel
    „description“: „string“,                 Beschreibung
    „startTime“: „datetime“,                 Beginn
    „endTime“: „datetime“,                   Ende
    „externalLink“: „string“,                Externer Link
    „learningLink“: „string“,                Learning Link
    „price“: „decimal“,                      Preis
    „examPrice“: „decimal“,                  Prüfungspreis
    „participantCount“: „int“,               Teilnehmerzahl
    „concept“: „string“,                     Konzept
    „instructors“: [„int“],                  Liste der Dozenten
    „categorys“: [„string“],                 Liste der Kategorien
    „meetings“: [{                           Liste der Termine
        „startTime“: „datetime“,                 Beginn
        „endTime“: „datetime“,                   Ende
        „hours“: „int“,                          Anzahl der Stunden
        „instructors“: [„int“]                   Liste der Dozenten
    }],
    „buildingName“: „string“,                Gebäudename
    „street“: „string“,                      Straße
    „city“: „string“,                        Stadt
    „zipCode“: „string“,                     Postleitzahl
    „totalHours“: „int“                      Total Hours
}
```

Beispiel: Übergabe von IDs

```
{
    „courseNumber“: „181-4B04“,
    „titel“: „Musterkurs“,
    „description“: „Musterbeschreibung“,
    „startTime“: „2020-02-28 08:00“,
    „endTime“: „2020-02-28 18:00“,
    „externalLink“: „Musterlink“,
    „learningLink“: „Musterlink“,
    „price“: 50.5,
    „examPrice“: 100,
    „participantCount“: 20,
    „concept“: „Musterkonzept“,
    „instructors“: [67, 204],
    „categorys“: [„Muster1“, „Muster2],
    „meetings“: [{
        „startTime“: „2020-02-28 08:00“,
        „endTime“: „2020-02-28 18:00“,
        „hours“: 10,
        „instructors“: [67]
    }],
    „buildingName“: „Mustergebäude“,
    „street“: „Musterstraße“,
    „city“: „Musterstadt“,
    „zipCode“: „Musterpostleitzahl“,
    „totalHours“: 10
}
```

Beispiel: Übergabe von Objekten

```
{
    „courseNumber“: „181-4B04“,
    „titel“: „Musterkurs“,
    „description“: „Musterbeschreibung“,
    „startTime“: „2020-02-28 08:00“,
    „endTime“: „2020-02-28 18:00“,
    „externalLink“: „Musterlink“,
    „learningLink“: „Musterlink“,
    „price“: 50.5,
    „examPrice“: 100,
    „participantCount“: 20,
    „concept“: „Musterkonzept“,
    „instructors“: [{
        „instructorNumber“: „67“,
        „gender“: „M“,
        „firstName“: „Max“,
        „lastName“: „Mustermann“,
        „email“: „Mustermail“,
        „job“: „Musterberuf“,
        „mobil“: „12345“,
        „tel“: „12345“,
        „fax“: „12345“
    }],
    „categorys“: [{
        „titel“: „Muster“,
        „description“: „Musterbeschreibung“
    }],
    „meetings“: [{
        „startTime“: „2020-02-28 08:00“,
        „endTime“: „2020-02-28 18:00“,
        „hours“: 10,
        „instructors“: [67]
    }],
    „buildingName“: „Mustergebäude“,
    „street“: „Musterstraße“,
    „city“: „Musterstadt“,
    „zipCode“: „Musterpostleitzahl“,
    „totalHours“: 10
}
```

Antwort:

Response Code 200: {

„courseNumber“: „string“,	Kursnummer
„titel“: „string“,	Titel
„description“: „string“,	Beschreibung
„uid“: „int“,	Eindeutige ID des Kurses
„startTime“: „datetime“,	Beginn
„endTime“: „datetime“,	Ende
„externalLink“: „string“,	Externer Link
„learningLink“: „string“,	Learning Link
„price“: „decimal“,	Preis
„examPrice“: „decimal“,	Prüfungspreis
„participantCount“: „int“,	Teilnehmerzahl
„concept“: „string“,	Konzept
„instructors“: [{	Liste der Dozenten
„instructorNumber“: „string“,	Dozentennummer
„gender“: „string“,	Anrede
„firstName“: „string“,	Vorname
„lastName“: „string“,	Nachname
„email“: „string“,	E-Mail
„job“: „string“,	Beruf
„mobil“: „string“,	Handynummer
„tel“: „string“,	Telefonnummer
„fax“: „string“,	Fax
„uid“: „int“	Eindeutige ID des Dozenten
}],	
„educationalInstruction“: {	Bildungseinrichtung
„name“: „string“,	Name
„street“: „string“,	Straße
„city“: „string“,	Stadt
„zipCode“: „string“,	Postleitzahl
„fallbackLink“: „string“,	Fallback Link
„tel“: „string“,	Telefonnummer
„logo“: „string“	Logo
},	
„categorys“: [{	Liste der Kategorien
„titel“: „string“,	Kategorietitel
„description“: „string“,	Kategoriebeschreibung
„uid“: „int“	Eindeutige ID der Kategorie
}],	

„meetings“: [{	Liste der Termine
“startTime“: „datetime“,	Beginn
„endTime“: „datetime“,	Ende
„hours“: „int“,	Anzahl der Stunden
„instructors“: [{	Liste der Dozenten des Termins
„instructorNumber“: „string“,	Dozentennummer
„gender“: „string“,	Anrede
„firstName“: „string“,	Vorname
„lastName“: „string“,	Nachname
„email“: „string“,	E-Mail
„job“: „string“,	Beruf
„mobil“: „string“,	Handynummer
„tel“: „string“,	Telefonnummer
„fax“: „string“,	Fax
„uid“: „int“	Eindeutige ID des Dozenten
}],	
„uid“: „int“	Eindeutige ID des Termins
}],	
„buildingName“: „string“,	Gebäudename
„street“: „string“,	Straße
„zipCode“: „string“,	Stadt
„city“: „string“,	Postleitzahl
„lat“: „decimal“,	Lat
„lng“: „decimal“,	Lng
„totalHours“: „int“	Gesamtzahl Stunden
}	

Response Code 404: Es existiert bereits ein Kurs / Keine Bildungseinrichtung für den ApiKey

OPTIONS	/rest/course/{apiKey}/update

Diese Abfrage soll nicht direkt von außen angesprochen werden. Die Abfrage ist dafür da, für die zugehörige POST-Methode die CORS-Richtlinien zu prüfen und wird dafür von modernen Browsern vor der POST-Abfrage aufgerufen.

POST	/rest/course/{apiKey}/update/uid

Aktualisiert einen Kurs für die im Parameter {apiKey} spezifizierte Bildungseinrichtung anhand der *uid*. Vor der Verwendung der Methode, muss es einen Applikationsschlüssel für die jeweilige Bildungseinrichtung geben. Dozenten und Termine können nur als Objekte angegeben werden. Ist eine *uid* enthalten, wird der das Objekt angepasst, sonst wird es erstellt und zugeordnet. Datumswerte können auf verschiedene Arten angegeben werden. Möglich sind die Schreibweise nach ISO oder DIN (Beispiele: 1970-01-01T00:00:00+00:00, 01.01.1970 00:00:00+00:00, …). Die Kategorien werden anhand von Schlagworten ausgewertet. Kommt ein Schlagwort in einer Kategorie vor, wird der Kurs zugeordnet. Alternativ können vollständige Kategorieobjekte übergeben werden. Pflichtfelder beim Anlegen sind die Kursnummer, der Titel sowie Beginn und Ende des Kurses.

Parameter:

```
{
    „courseNumber“: „string“,                  Kursnummer
    „titel“: „string“,                         Titel
    „description“: „string“,                   Beschreibung
    „startTime“: „datetime“,                   Beginn
    „endTime“: „datetime“,                     Ende
    „externalLink“: „string“,                  Externer Link
    „learningLink“: „string“,                  Learning Link
    „price“: „decimal“,                        Preis
    „examPrice“: „decimal“,                    Prüfungspreis
    „participantCount: „int“,                  Teilnehmerzahl
    „concept“: „string“,                       Konzept
    „instructors“: [{                          Liste der Dozenten
        „instructorNumber“: „string“,              Dozentennummer
        „gender“: „string“,                        Anrede/Geschlecht
        „firstName“: „string“,                     Vorname
        „lastName“: „string“,                      Nachname
        „email“: „string“,                         E-Mail
        „job“: „string“,                           Beruf
        „mobil“: „string“,                         Mobilnummer
        „tel“: „string“,                           Telefonnummer
        „fax“: „string“,                           Fax
        „uid“: „int“                               Uid
    }],
    „categorys“: [„string“],                   Liste der Kategorien
    „meetings“: [{                             Liste der Termine
        „startTime“: „datetime“,                   Beginn
        „endTime“: „datetime“,                     Ende
        „hours“: „int“,                            Anzahl der Stunden
        „instructors“: [„instructor“]              Liste der Dozenten
    }],
    „buildingName“: „string“,                  Gebäudename
    „street“: „string“,                        Straße
    „city“: „string“,                          Stadt
    „zipCode“: „string“,                       Postleitzahl
    „totalHours“: „int“                        Total Hours
}
```

Beispiel:


```
{
    „courseNumber“: „181-4B04“,
    „titel“: „Musterkurs“,
    „description“: „Musterbeschreibung“,
    „startTime“: „2020-02-28 08:00“,
    „endTime“: „2020-02-28 18:00“,
    „externalLink“: „Musterlink“,
    „learningLink“: „Musterlink“,
    „price“: 50.5,
    „examPrice“: 100,
    „participantCount: 20,
    „concept“: „Musterkonzept“,
    „instructors“: [{
        „instructorNumber“: „67“,
        „gender“: „M“,
        „firstName“: „Max“,
        „lastName“: „Mustermann“,
        „email“: „Mustermail“,
        „job“: „Musterberuf“,
        „mobil“: „12345“,
        „tel“: „12345“,
        „fax“: „12345“,
        „uid“: 123
    }],
    „categorys“: [„Muster1“, „Muster2“],
    „meetings“: [{
        „startTime“: „2020-02-28 08:00“,
        „endTime“: „2020-02-28 18:00“,
        „hours“:
        „instructors“: [[{
            „instructorNumber“: „67“,
            „gender“: „M“,
            „firstName“: „Max“,
            „lastName“: „Mustermann“,
            „email“: „Mustermail“,
            „job“: „Musterberuf“,
            „mobil“: „12345“,
            „tel“: „12345“,
            „fax“: „12345“,
            „uid“: 123
        }]
    }],
    „buildingName“: „Mustergebäude“,
    „street“: „Musterstraße“,
    „city“: „Musterstadt“,
    „zipCode“: „Musterpostleitzahl“,
    „totalHours“: 10
}
```

Antwort:

Response Code 200: {

„courseNumber“: „string“,	Kursnummer
„titel“: „string“,	Titel
„description“: „string“,	Beschreibung
„uid“: „int“,	Eindeutige ID des Kurses
„startTime“: „datetime“,	Beginn
„endTime“: „datetime“,	Ende
„externalLink“: „string“,	Externer Link
„learningLink“: „string“,	Learning Link
„price“: „decimal“,	Preis
„examPrice“: „decimal“,	Prüfungspreis
„participantCount“: „int“,	Teilnehmerzahl
„concept“: „string“,	Konzept
„instructors“: [{	Liste der Dozenten
„instructorNumber“: „string“,	Dozentennummer
„gender“: „string“,	Anrede
„firstName“: „string“,	Vorname
„lastName“: „string“,	Nachname
„email“: „string“,	E-Mail
„job“: „string“,	Beruf
„mobil“: „string“,	Handynummer
„tel“: „string“,	Telefonnummer
„fax“: „string“,	Fax
„uid“: „int“	Eindeutige ID des Dozenten
}],	
„educationalInstruction“: {	Bildungseinrichtung
„name“: „string“,	Name
„street“: „string“,	Straße
„city“: „string“,	Stadt
„zipCode“: „string“,	Postleitzahl
„fallbackLink“: „string“,	Fallback Link
„tel“: „string“,	Telefonnummer
„logo“: „string“	Logo
},	
„categorys“: [{	Liste der Kategorien
„titel“: „string“,	Kategorietitel
„description“: „string“,	Kategoriebeschreibung
„uid“: „int“	Eindeutige ID der Kategorie
}],	

„meetings“: [{	Liste der Termine
“startTime“: „datetime“,	Beginn
„endTime“: „datetime“,	Ende
„hours“: „int“,	Anzahl der Stunden
„instructors“: [{	Liste der Dozenten des Termins
„instructorNumber“: „string“,	Dozentennummer
„gender“: „string“,	Anrede
„firstName“: „string“,	Vorname
„lastName“: „string“,	Nachname
„email“: „string“,	E-Mail
„job“: „string“,	Beruf
„mobil“: „string“,	Handynummer
„tel“: „string“,	Telefonnummer
„fax“: „string“,	Fax
„uid“: „int“	Eindeutige ID des Dozenten
}],	
„uid“: „int“	Eindeutige ID des Termins
}],	
„buildingName“: „string“,	Gebäudename
„street“: „string“,	Straße
„zipCode“: „string“,	Stadt
„city“: „string“,	Postleitzahl
„lat“: „decimal“,	Lat
„lng“: „decimal“,	Lng
„totalHours“: „int“	Gesamtzahl Stunden
}	

Response Code 404: Es existiert bereits ein Kurs / Keine Bildungseinrichtung für den ApiKey

OPTIONS	/rest/course/{apiKey}/update/uid

Diese Abfrage soll nicht direkt von außen angesprochen werden. Die Abfrage ist dafür da, für die zugehörige POST-Methode die CORS-Richtlinien zu prüfen und wird dafür von modernen Browsern vor der POST-Abfrage aufgerufen.

POST	/rest/course/filter

Durchsucht die Kurse anhand des übergebenen Filters. Hierbei können verschiedene Angaben getätigt werden. Ist nur ein Suchstring angegeben, werden alle Kurse gesucht, in denen der Suchstring der Kursnummer entspricht. Alternativ kann der Suchstring auch im Titel oder der Beschreibung des Kurses enthalten sein. Ist eine ID angegeben, werden die Kurse zu der Kategorie mit der angegebenen ID gesucht. Ist ein Filter angegeben, wird basierend auf diesem Filterobjekt gesucht. Es ist hierbei auch möglich, Kombinationen aller Suchmöglichkeiten anzuwenden. Wichtig bei der Angabe von Filtern ist, welche Angaben zusammenhängen:

- Ist die Positionssuche aktiviert, müssen auch die Breiten- und Längengrade angegeben werden.
- Ist die Preissuche aktiviert, müssen auch die untere und obere Grenze angegeben werden.
- Ist die Suche nach einer Bildungseinrichtung aktiviert, muss auch mindestens eine ID in der Liste mitgegeben werden.
- Ist die Datumssuche aktiviert, müssen auch ein Start- und ein Enddatum angegeben werden.

Parameter:

```
{
    „searchString“: „string“,                       Suchstring
    „categoryUid“: „int“,                           Eindeutige ID der zu durchsuchenden Kategorie
    „filter“: {                                     Filterobjekt
        „radiusEnabled“: boolean,                       Positionssuche aktivieren
        „location“: {                                   Positionsfilter
            „lat“: „decimal“,                               Lat
            „lng“: „decimal“                                Lng
        },
        „priceEnabled“: „boolean“,                      Preissuche aktivieren
        „price“: {                                      Preisfilter
            „lower“: „decimal“,                             Untere Preisgrenze
            „upper“: „decimal“                              Obere Preisgrenze
        },
        „educationEnabled“: „boolean“,                  Bildungseinrichtungssuche aktivieren
        „activEducationalInstructions“: [„int“],        Liste der IDs der Bildungseinrichtungen
        „dateEnabled“: „boolean“,                       Datumssuche aktivieren
        „start“: „datetime“,                            Startzeitpunkt
        „end“: „datetime“                               Endzeitpunkt
    }
}
```

Beispiel: Suche mit Suchstring

```
{
    „searchString“: „181-4B04“
}
```

Beispiel: Suche nach Kategorie

```
{
    „categoryUid“: 1
}
```

Beispiel: Suche mit Positionsfilter

```
{
    „filter“: {
        „radiusEnabled“: true,
        „location“: {
            „lat“: 45,
            „lng“: 45
        }
    }
}
```

Beispiel: Suche mit Preisfilter

```
{
    „filter“: {
        „priceEnabled“: true,
        „price“: {
            „lower“: 0,
            „upper“: 100
        }
    }
}
```

Beispiel: Suche mit Datumsfilter

```
{
    „filter“: {
        „dateEnabled“: true,
        „start“: "1970-01-01T00:00“,
        „end“: „1970-01-01Z00:00“
    }
}
```

Beispiel: Suche nach Bildungseinrichtungen

```
{
    „filter“: {
        „educationEnabled“: true,
        „activEducationalInstructions“: [1]
    }
}
```

Beispiel: Kombinierte Suche 1

```
{
    „searchString“: „181-4B04“,
    „filter“: {
        „priceEnabled“: true,
        „price“: {
            „lower“: 0,
            „upper“: 100
        }
    }
}
```

Beispiel: Kombinierte Suche 2

```
{
    „categoryUid“: 1,
    „filter“: {
        „educationEnabled“: true,
        „activEducationalInstructions“: [1],
        „dateEnabled“: true,
        „start“: "1970-01-01T00:00“,
        „end“: „1970-01-01Z00:00“
    }
}
```

Antwort:

Response Code 200: [{

„courseNumber“: „string“,	Kursnummer
„titel“: „string“,	Titel
„description“: „string“,	Beschreibung
„uid“: „int“	Eindeutige ID des Kurses

}]

OPTIONS	/rest/course/filter
Diese Abfrage soll nicht direkt von außen angesprochen werden. Die Abfrage ist dafür da, für die zugehörige POST-Methode die CORS-Richtlinien zu prüfen und wird dafür von modernen Browsern vor der POST-Abfrage aufgerufen.	
DELETE	/rest/course/{apiKey}/{courseNumber}
Löscht den Kurs mit der im Parameter {courseNumber} übergebenen Kursnummer für die über den Parameter {apiKey} spezifizierte Bildungseinrichtung. Außerdem werden die Termine gelöscht, die dem Kurs zugeordnet sind.	
Parameter:	
apiKey = „string“, courseNumber = „string“	Applikationsschlüssel der Bildungseinrichtung Kursnummer
Beispiel:	
apiKey = „Mustereinreichtung“, courseNumber = „181-4B04“	
Antwort:	
Response Code 200: - **Response Code 404**: Es gibt keinen Kurs zur Nummer / Keine Bildungseinrichtung für den ApiKey	
OPTIONS	/rest/course/{apiKey}/{courseNumber}
Diese Abfrage soll nicht direkt von außen angesprochen werden. Die Abfrage ist dafür da, für die zugehörige DELETE-Methode die CORS-Richtlinien zu prüfen und wird dafür von modernen Browsern vor der DELETE -Abfrage aufgerufen.	
DELETE	/rest/course/{uid}
Löscht den Kurs mit der im Parameter {uid} übergebenen. Außerdem werden die Termine gelöscht, die dem Kurs zugeordnet sind.	
Parameter:	
apiKey = „string“, courseNumber = „string“	Applikationsschlüssel der Bildungseinrichtung Kursnummer
Beispiel:	
apiKey = „Mustereinreichtung“, courseNumber = „181-4B04“	
Antwort:	
Response Code 200: - **Response Code 404**: Es gibt keinen Kurs zur Nummer / Keine Bildungseinrichtung für den ApiKey	
OPTIONS	/rest/course/{uid}
Diese Abfrage soll nicht direkt von außen angesprochen werden. Die Abfrage ist dafür da, für die zugehörige DELETE-Methode die CORS-Richtlinien zu prüfen und wird dafür von modernen Browsern vor der DELETE -Abfrage aufgerufen.	

EducationalInstruction

GET	/rest/educationalInstruction
Gibt alle Bildungseinrichtungen zurück.	
Parameter: -	
Beispiel: -	
Antwort: **Response Code 200**: [{ „name“: „string“, „uid“: „int“ }]	 Name Eindeutige ID der Bildungseinrichtung

GET	/rest/educationalInstruction/{apiKey}
Gibt den Name und die eindeutige ID für die über den Parameter {apiKey} spezifizierte Bildungseinrichtung zurück.	
Parameter: apiKey = „string“	 Applikationsschlüssel der Bildungseinrichtung
Beispiel: apiKey = „Musterkey“	

Antwort:
 	
Response Code 200: {
„name“: „string“,
„uid“: „int“
}
	
	Name
	Eindeutige ID der Bildungseinrichtung

Category

GET	/rest/category
Gibt alle Kategorien zurück.	
Parameter: -	
Beispiel: -	
Antwort:	
Response Code 200: [{	
„titel“: „string“,	Titel
„description“: „string“,	Beschreibung
„uid“: „int“	Eindeutige ID der Kategorie
}]	

Meeting

GET	/rest/meeting

Gibt alle Termine mit Kursinformationen zurück.

Parameter: -

Beispiel: -

Antwort:

Response Code 200: [{

„titel“: „string“,	Titel des Kurses
„courseNumber“: „string“,	Kursnummer
„startTime“: „datetime“,	Startzeit des Termins
„endTime“: „datetime“,	Endzeit des Termins
„instructor“: „string“,	Ein Dozent des Termins
„city“, „string“	Ort

}]

GET	/rest/meeting/educationalInstruction/{uid}

Gibt die Termine mit Kursinformationen zurück, die zur im Parameter {uid} spezifizierten Bildungseinrichtung gehören.

Parameter:

uid = „int“	Eindeutige ID der Bildungseinrichtung

Beispiel:

uid = 1

Antwort:

Response Code 200: [{

„titel“: „string“,	Titel des Kurses
„courseNumber“: „string“,	Kursnummer
„startTime“: „datetime“,	Startzeit des Termins
„endTime“: „datetime“,	Endzeit des Termins
„instructor“: „string“,	Ein Dozent des Termins
„city“, „string“	Ort

}]

Instructor

GET	/rest/instructor/educationalInstruction/{uid}

Gibt alle Dozenten zur Bildungseinrichtung mit der angegebenen ID im Parameter {uid} zurück.

Parameter:

uid: „int“	Eindeutige ID der Bildungseinrichtung

Beispiel:

uid = 1

Antwort:

Response Code 200: [{

„instructorNumber“: „string“,	Dozentennummer
„gender“: „string“,	Anrede
„firstName“: „string“,	Vorname
„lastName“: „string“,	Nachname
„email“: „string“,	E-Mail
„job“: „string“,	Beruf
„tel“: „string“,	Handynummer
„mobil“: „string“,	Telefonnummer
„fax“: „string“,	Fax
„uid“: „int“	Eindeutige ID des Dozenten

}]

POST	/rest/instructor/{apiKey}/create

Erstellt einen Dozenten für die über den Parameter {apiKey} spezifizierte Bildungseinrichtung. Bevor diese Methode verwendet werden kann, muss zuerst ein Applikationsschlüssel für die jeweilige Bildungseinrichtung hinterlegt werden. Wichtig beim Anlegen eines Dozenten ist hierbei, dass die Dozentennummer entweder leer oder eindeutig sein muss. Das bedeutet, dass ein Dozent keine Dozentennummer haben kann, allerdings auch keine zwei Dozenten die gleiche Nummer aufweisen dürfen.

Parameter:

{

„instructorNumber“: „string“,	Dozentennummer
„gender“: „string“,	Anrede
„firstName“: „string“,	Vorname
„lastName“: „string“,	Nachname
„email“: „string“,	E-Mail
„job“: „string“,	Beruf
„tel“: „string“,	Handynummer
„mobil“: „string“,	Telefonnummer
„fax“: „string“	Fax

}

Beispiel:

```
{
    „instructorNumber“: „40“,
    „gender“: „Herr“,
    „firstName“: „Max“,
    „lastName“: „Mustermann“,
    „email“: „max.mustermann@muster.de“,
    „job“: „Musterjob“,
    „tel“: „012345“,
    „mobil“: „012345“,
    „fax“: „012345“
}
```

Antwort:

Response Code 200: [{

„instructorNumber“: „string“,	Dozentennummer
„gender“: „string“,	Anrede
„firstName“: „string“,	Vorname
„lastName“: „string“,	Nachname
„email“: „string“,	E-Mail
„job“: „string“,	Beruf
„tel“: „string“,	Handynummer
„mobil“: „string“,	Telefonnummer
„fax“: „string“,	Fax
„uid“: „int“	Eindeutige ID des Dozenten

}]

Response Code 404: Bereits vorhandene Dozentennummer oder ungültiger Key

OPTIONS	/rest/instructor/{apiKey}/create

Diese Abfrage soll nicht direkt von außen angesprochen werden. Die Abfrage ist dafür da, für die zugehörige POST-Methode die CORS-Richtlinien zu prüfen und wird dafür von modernen Browsern vor der POST-Abfrage aufgerufen.

Mail

GET	/rest/mail/key

Holt einen Schlüssel zum Versenden einer E-Mail vom Server. Dieser muss der Methode *send* zum Senden einer E-Mail übergeben werden und ist eine halbe Stunde gültig.

Parameter: -

Beispiel: -

Antwort:

Response Code 200: {
„key“: „string“,
}

POST	/rest/mail/send

Sendet eine E-Mail an info@erwachsenenbildung-lsa.de und übermittelt dabei die im Parameter angegebenen Werte. Hierbei ist ein gültiger Key zu übergeben, der über die Methode *key* geholt werden kann.

Parameter:

{

„key“: „string“,	Schlüssel als Base64-String
„name“: „string“,	Vor- und Nachname
„company“: „string“,	Firma
„companyphone“: „string“,	Firma Telefonnummer
„phone“: „string“,	Telefonnummer
„email“: „string“,	E-Mail
„subject“: „string“,	Thema
„content“: „string“,	Inhalt

}

Beispiel:

{
„key“: „MTU5NjQ1Mzc4Ng==“,
„name“: „Max Mustermann“,
„company“: „Musterfirma“,
„companyphone“: „+49 12345“,
„phone“: „+49 56789“,
„email“: „Mustermann@Mustermail.com“,
„subject“: „Musterthema“,
„content“: „Musterinhalt“,
}

Antwort:

Response Code 200: Die E-Mail wurde erfolgreich versendet
Response Code 404: Fehlende Eingaben oder ungültiger Key