

# **PAUKR Rest API Dokumentation**

Stand 04.11.2021

Autor:

Sebastian Beyl

## Grundsätzliches

Die PAUKR API ist eine klassische REST API auf Basis von HTTPS. Alle Anfragen werden als HTTPS-Requests gestellt, die Kommunikation läuft über den Request bzw. als JSON-codierte Übertragung. Das folgende Dokument beschreibt die Möglichkeiten der API mit dem jeweiligen Request und dem zu erwartenden Response. Unabhängig vom Request hat immer eine Autorisierung bei jeder Übertragung zu erfolgen.

## Autorisierung

Die Authorisation einer Anfrage erfolgt über einen Bearer Token. Um diesen zu senden, wird in jeder Anfrage ein HTTP-Header mitgesendet, der folgenden Inhalt hat:

```
Authorization: Bearer XXX
```

XXX steht dabei für den Token, der vom Zentrum für LehrerInnenbildung für jedes beteiligte Projekt ausgestellt wird. Bitte verwenden Sie immer nur den Token, der ihrem Projekt zugeteilt ist! Nur so ist gewährleistet, dass wir eventuell missbräuchlich genutzte Software deaktivieren, ohne andere Software oder Dienste in Mitleidenschaft zu ziehen.

Der Token muss bei jeder Anfrage, egal ob der GET oder POST gestellt, erneut gesendet werden. Eine sessionbasierte Autorisierung ist nicht möglich.

## Aufbau einer Anfrage

Eine Anfrage wird per GET oder POST gestellt, gemäß der Logik einer REST API. Die Kommunikation erfolgt im Regelfall per JSON. Wird innerhalb des POST-Requests JSON übergeben, muss der Content-Type im Header entsprechend gesetzt werden.

```
Content-Type:application/json
```

## Aufbau einer Antwort

Unabhängig von der Übertragungsart wird die Antwort der API in JSON zurückgegeben. Die Felder variieren je nach Aufruf und werden in dieser Dokumentation näher erläutert. Allerdings sind einige Felder in jeder Antwort vorhanden. Die Minimal-Antwort einer erfolgreichen Antwort sieht folgendermaßen aus:

```
{
  "status": "success",
  "statusCode": 200,
  "statustext": "successful request"
}
```

Der zurückgegebene HTTP-Response Code ist 200 und entspricht damit dem statuscode der übergebenen Antwort. Eine nicht erfolgreiche Antwort enthält minimal die folgenden Felder:

```
{
  "status": "error",
  "statuscode": 401,
  "statustext": "Authorization failed"
}
```

Der zurückgegebene HTTP-Response Code entspricht dem angezeigten statuscode in der Antwort.

## Die API Requests

### GET POST /api/v1/

Die API kann testweise ohne Übermittlung von Parametern aufgerufen werden. In diesem Fall wird lediglich die Autorisierung überprüft. Der Request kann wahlweise per GET oder POST erfolgen. Ist die Autorisierung gültig, antwortet die API mit folgender Antwort:

```
{
  "status": "success",
  "statuscode": 200,
  "statustext": "successful request"
}
```

Ist die Autorisierung hingegen fehlgeschlagen, antwortet die API mit folgendem Response:

```
{
  "status": "error",
  "statuscode": 401,
  "statustext": "Authorization failed"
}
```

Auf diese Art und Weise kann die eigene Autorisierung überprüft werden, aber auch die Funktion bzw. Verfügbarkeit der API, ohne dass eine Autorisierung mitgegeben werden muss.

### GET /api/v1/matrikels/xxxxxxx/papers

Der Request wird ohne weitere Parameter gestellt. Der Platzhalter „xxxxxxx“ steht für die Matrikelnummer des Studierenden, dessen Aktendokumente angezeigt werden sollen. Bei einer erfolgreichen Anfrage gibt die API alle verfügbaren Dokumente des Studierenden wieder, dessen Matrikelnummer in der Anfrage übergeben wurde.

Die Matrikelnummer muss vollständig angegeben werden und genau sieben Ziffern beinhalten. Ist das nicht der Fall, wird die Anfrage scheitern. Ist keine Akte des Studierenden vorhanden, wird die Anfrage nicht scheitern, sondern erfolgreich, aber mit einem leeren Ergebnis beantwortet.

Ein erfolgreicher Response sieht wie folgt aus:

```
{
  "status": "success",
  "statusCode": 200,
  "statustext": "successful request",
  "papers": [
    "20161018_BA_ErstGutA.pdf",
    "20161025_BA_ZweitGutA.pdf",
    "20170727_BA_Zeugnisantrag.pdf"
  ]
}
```

Innerhalb des Response wird das Array „papers“ übergeben, dessen Inhalt die zu der Matrikelnummer verfügbaren Akteneinträge beinhaltet. Diese Akteneinträge können dann aufgerufen werden.

Der Response „papers“ beinhaltet bei erfolgreicher Antwort immer ein Array, ist keine Akte vorhanden, ist dieses Array ohne Elemente, wird aber dennoch ausgegeben.

## GET /api/v1/matrikels/xxxxxxx/papers/xxx.pdf

Der Request wird ohne weitere Parameter gestellt. Der Platzhalter „xxxxxxx“ steht für die Matrikelnummer des Studierenden, der Platzhalter xxx.pdf für das Dokument, das aus der Akte abgerufen werden soll. Dieses Dokument muss existieren, eine Abfrage der verfügbaren Dokumente zu einer Matrikelnummer ist ohne Angabe des Dokuments möglich.

```
{
  "status": "success",
  "statusCode": 200,
  "statustext": "successful request",
  "matrikel": "1234567",
  "paper": "19771208_MA_Abschlussarbeit.pdf",
  "file": "file content as base64 encoded string"
}
```

Die Parameter matrikel und paper werden noch einmal, obwohl im Request gestellt, im Response angezeigt. Der Inhalt der Datei, der im Regelfall als binary vorliegt, wird als base64-codierter String in der Variablen „file“ übertragen. Um die binären Daten wiederherzustellen, kann der String einfach wieder base64-decodiert werden.

## POST /api/v1/matrikels/xxxxxxx/papers

Dieser Request wird gestellt, um ein neues Dokument in die Akte des Studierenden zu legen. Dabei sind eine ganze Reihe Regeln zu beachten, denn zum einen müssen eine Reihe Metadaten als Parameter übertragen werden, zum anderen dürfen diese Parameter nur ein festes Set an Daten beinhalten. Dieses Verhalten wird im folgenden beschrieben.

Der Platzhalter „xxxxxxx“ innerhalb der URL steht für die Matrikelnummer des Studierenden, in dessen Akte ein neues Dokument eingefügt werden soll. Diese Matrikelnummer darf nur aus Ziffern

bestehen, aber mit einer „0“ beginnen. Alle weiteren Parameter werden innerhalb des POST-Requests als JSON übergeben:

```
{
  "date": "19770812",
  "studgrad": "MA",
  "type": "Zeugnisantrag",
  "filetype": "pdf",
  "file": "base 64 encoded (binary) file content"
}
```

Die einzelnen Parameter sind sehr sensitiv gegen Veränderungen. Daher ist hier auf eine sehr genaue Bearbeitung zu achten, im Regelfall wird auch Groß- und Kleinschreibung unterschieden.

Name des Parameters	Bedeutung	Erlaubte Werte
date	0	0
studgrad	0	MA BA
type	0	AbArZulassung Abschlussarbeit Anerkannt DrittGutA ErstGutA Zeugnisantrag ZweitGutA
filetype	0	pdf
file	Daten (auch binär) des Files als base64 codierter String	base64 encoded string

Achtung! Ein Dokument der Akte kann NICHT überschrieben werden. Ist ein Dokument einmal in die Akte eingefügt, kann es über die API nicht mehr überschrieben oder gelöscht werden.

Der Response gibt der Erfolg oder einen Fehler zurück. Im Regelfall gibt der statustext Auskunft über den genauen Grund des Fehlers.

```
{
  "status": "success",
  "statusCode": 200,
  "statustext": "success"
}
```