

# Dokumentation: Datenbank / API der Pflanzenbestimmungssoftware

Steven-Alexander Maack – Stand: 01.10.2020

## Inhaltsverzeichnis:

1. [Vorwort](#)
2. [Die Datenbank - Pflanzenbestimmung](#)
  - 2.1. [Legende DB](#)
  - 2.2. [Tabelle: abgefragt](#)
  - 2.3. [Tabelle: admins](#)
  - 2.4. [Tabelle: ausbildungsart](#)
  - 2.5. [Tabelle: azubis](#)
  - 2.6. [Tabelle: fachrichtung](#)
  - 2.7. [Tabelle: pflanze](#)
    - 2.7.1. [Tabelle: p antworten](#)
    - 2.7.2. [Tabelle: p bilder](#)
    - 2.7.3. [Tabelle: p kategorien](#)
  - 2.8. [Tabelle: quiz art](#)
    - 2.8.1. [Tabelle: quiz p zuweisung](#)
  - 2.9. [Tabelle: statistik](#)
    - 2.9.1. [Tabelle: stat einzel](#)
    - 2.9.2. [Tabelle: stat einzel detail](#)
3. [Die Schnittstelle](#)
  - 3.1. [Legende API](#)
  - 3.2. [Login](#)
  - 3.3. [Admins](#)
  - 3.4. [Azubis](#)
    - 3.4.1. [Ausbildungsart](#)
    - 3.4.2. [Fachrichtung](#)
  - 3.5. [Pflanzen](#)
    - 3.5.1. [Abgefragt](#)
    - 3.5.2. [Bilder](#)
    - 3.5.3. [Kategorien](#)
  - 3.6. [Quiz](#)
    - 3.6.1. Art
    - 3.6.2. Pflanzen Zuweisung
  - 3.7. [Statistik](#)

# 1. Vorwort

Diese Dokumentation dient der Orientierung und Hilfe zur Nutzung der Datenbank und API für die TSBW Interne „Pflanzenbestimmung“ – Lernsoftware.

Aktuell ist diese nur aufgrund oben genanntem grob ausgearbeitet und wird eventuell zu einem späteren Zeitpunkt überarbeitet. (01.10.2020)

## 2. Die Datenbank - Pflanzenbestimmung

### 2.1. Legende:

AI: Auto Increment, automatische Anhebung von indiziertem Wert bei neuem Eintrag, z.B. der ID.

BLOB: Binary Large Object, kann variable große Daten enthalten.

INT: Integer, ein numerischer Wert.

TINYINT: Kleiner numerischer Wert, z.B.: 1 oder 0.

VARCHAR: Eine Zeichenkette aka.: Ein String.

N: Null, nicht Zuweisung von Wert zulässig.

NN: Not Null, nicht-Zuweisung von Wert unzulässig.

UN: Unsigned, numerischer Wert ohne Vorzeichen, z.B.: - 123

UQ: Unique, einzigartiger Eintrag. Z.B ID = 1, darf nur einmal existieren.

PK: Primary Key, der Referenz-Schlüssel der Tabelle

FK: Foreign Key, der Fremdschlüssel einer anderen Tabelle.

On Delete Cascade: Wenn der Referenzierte Eintrag gelöscht wird, so wird auch dieser Eintrag gelöscht.

On Update Cascade: Wenn der Referenzierte Eintrag geändert wird, erhält auch dieser Eintrag den neuen Wert.

## 2.2: Tabelle: abgefragt

**Engine:** InnoDB

**Default Charakter Set** = utf8;

Dient zur Lagerung von bereits abgefragten Pflanzen innerhalb der Lernsoftware, der Zählung von Erfolgen in Folge.

Wenn sieben Mal in Folge eine Pflanze richtig beantwortet wurde, soll diese als „gelernt“ abgespeichert werden.

Table: **abgefragt**

**Columns:**

<b>fk_azubi</b>	int(10) UN
<b>fk_pflanze</b>	int(10) UN
<b>counter_korrekt</b>	int(10) UN
<b>gelernt</b>	tinyint(3) UN

**Enthält folgende Felder:**

*fk\_azubi* – INT, UN, NN

Enthält den Fremdschlüssel des zugewiesenen Nutzers.

*fk\_pflanze* – INT, UN, NN

Enthält den Fremdschlüssel der abgefragten Pflanze.

*counter\_korrekt* – INT UN NN

Enthält Anzahl der Erfolgreich beantworteten selben Pflanze in Folge.

*gelernt* – TINYINT, UN, NN

Dient zur Abfrage, ob die Pflanze schon ausreichend gelernt wurde.

**Primärschlüssel:** fk\_azubi, fk\_pflanze

**Constraints:**

1. *azubi-gelernt*

Fremdschlüssel: fk\_azubi

Referenz: azubis - id

Verhalten: On Delete Cascade, On Update Cascade.

2. *pflanze-gelernt*

Fremdschlüssel: fk\_pflanze

Referenz: planze - id

Verhalten: On Delete Cascade, On Update Cascade.

## 2. 3: Tabelle: admins

**Engine:** InnoDB

**Default Charakter Set** = utf8;

Dient der Lagerung und Verwaltung von Admin/Ausbilder-Nutzerdaten.

Es sollte von vorn herein mindestens ein System-Admin mit der Berechtigungsflag von 1, als Super User existieren.

<b>Table: admins</b>	
<b>Columns:</b>	
<b>id</b>	int(10) UN AI PK
<b>nutzernamen</b>	varchar(45)
vorname	varchar(45)
name	varchar(45)
passwort	varchar(64)
berflag	int(11)

### Enthält folgende Felder:

*id* - INT, UN, AI, NN

*nutzernamen* - VARCHAR (45), NN, UQ

*vorname* - VARCHAR (45), NN

*name* - VARCHAR (45), NN

*passwort* - VARCHAR (64), NN

*berflag* – INT, NN, DEFAULT 2

„berflag“(Berechtigungsflag) dient zur Verteilung von Zugriffsberechtigungen, wobei jedem Ausbilder bei Erschaffung der Wert „2“ zugewiesen wird.

Darüber existiert noch der Wert 1, welcher den System Admins reserviert ist.  
Ein System Admin mit der Flag 1 ist z.B. in der Lage andere Admins zu löschen,

was ein Admin mit der Flag 2 nicht können soll.

**Primärschlüssel:** *id*

## 2.4: Tabelle: ausbildungsart

**Engine:** InnoDB

**Default Charakter Set** = utf8;

Dient zur Lagerung der vordefinierten Ausbildungsarten „Vollzeit“ und „Werker“.

Auf diesen Ausbildungsarten basiert, welche Kategorien bei der Auswertung gewertet werden und welche nicht.

### Enthält folgende Felder:

*id* – INT, UN, NN, AI

*name* – VARCHAR (45)

**Primärschlüssel:** id

<b>Table:</b> ausbildungsart
<b>Columns:</b>
<b>id</b> int(10) UN AI PK
name varchar(45)

## 2.5: Tabelle: azubis

**Engine:** InnoDB

**Default Charakter Set** = utf8;

Dient der Lagerung und Verwaltung von Teilnehmer/Azubi-Nutzerdaten, des zugewiesenen Ausbilders, der Fachrichtung, der Quiz-Art, und ob der Nutzer im Prüfungsmodus ist oder nicht.

Table: **azubis**

**Columns:**

<b>id</b>	int(10) UN AI PK
<b>fk_ausbilder</b>	int(10) UN
<b>fk_ausbildungsart</b>	int(10) UN
<b>fk_fachrichtung</b>	int(10) UN
<b>fk_quiz_art</b>	int(10) UN
<b>nutzername</b>	varchar(45)
<b>passwort</b>	varchar(64)
<b>name</b>	varchar(45)
<b>vorname</b>	varchar(45)
<b>pruefung</b>	tinyint(3) UN

**Enthält folgende Felder:**

*fk\_ausbilder* - INT, UN, DEFAULT 1

Enthält den Fremdschlüssel des zugewiesenen Ausbilders/Admins.

*fk\_ausbildungsart* - INT, UN

Enthält den Fremdschlüssel der des Nutzers zugewiesenen Ausbildungsart.

*fk\_fachrichtung* - INT, UN

Enthält den Fremdschlüssel der des Nutzers zugewiesenen Fachrichtung.

*fk\_quiz\_art* - INT, UN, DEFAULT 0

Enthält den Fremdschlüssel der des Nutzers zugewiesenen Quiz-Art/Größe

*id* - INT, UN, AI, NN

*nutzername* - VARCHAR (45), NN

*passwort* - VARCHAR (64), NN

*name* – VARCHAR (45), NN

*vorname* – VARCHAR (45), NN

*pruefung* – TINYINT, UN, DEFAULT 0

Nutzer im Prüfungsmodus ist oder nicht. Z.B.: 1 oder 0.

**Primärschlüssel:** *id*

**Constraints:**

**1. ausbilder**

Fremdschlüssel: *fk\_ausbilder*

Referenz: azubis - *id*

Verhalten: On Update Cascade

**Anmerkung:** Um die Referenzielle Integrität intakt zu halten, sollte bei Löschung eines Admins der Fremdschlüssel neu zugewiesen werden!

## 2. user-ausbildungsart

Fremdschlüssel: fk\_ausbildungsart

Referenz: ausbildungsart - id

Verhalten: On Update Cascade

## 3. user-quiz\_art

Fremdschlüssel: fk\_ausbildungsart

Referenz: quiz\_art - id

Verhalten: On Update Cascade

## 4. user-richtung

Fremdschlüssel: fk\_fachrichtung

Referenz: fachrichtung - id

Verhalten: On Update Cascade

## 2.6: Tabelle: fachrichtung

**Engine:** InnoDB

**Default Charakter Set** = utf8;

Dient zur Lagerung der aktuellen Bezeichnungen der Fachrichtungen.

### Enthält folgende Felder:

*id* – INT, UN, NN, UQ, AI

*name* – VARCHAR (45), NN, UQ

**Primärschlüssel:** *id*

**Table:** fachrichtung

**Columns:**

<b>id</b>	int(10) UN AI PK
name	varchar(45)

## 2.7: Tabelle: pflanze

**Engine:** InnoDB

**Default Charakter Set** = utf8;

Dient zur Lagerung der Pflanzen ID's und ihren Fachrichtungs-Zuweisungen

**Table:** pflanze

**Columns:**

<b>id</b>	int(10) UN AI PK
zierbau	tinyint(3) UN
galabau	tinyint(3) UN

### Enthält folgende Felder:

*id* – INT, UN, NN, UQ, AI

*zierbau* – TINYINT, UN, NN

Enthält ob die Pflanze zu den Zierbaupflanzen gehört oder nicht.

*galabau* – TINYINT, UN, NN

Enthält ob die Pflanze zu den Garten- und Landschaftsbaupflanzen gehört oder nicht.

**Primärschlüssel:** *id*



### 2.7.1: Tabelle: p\_antworten

**Engine:** InnoDB

**Default Charakter Set** = utf8;

Dient zur Lagerung der möglichen Antworten in Bezug auf die hinterlegten Kategorien.

Table: **p\_antworten**

Columns:

<u>fk_pflanze</u>	int(10) UN PK
<u>fk_kategorie</u>	int(10) UN PK
antwort	varchar(45)

#### **Enthält folgende Felder:**

*fk\_pflanze* – INT, UN, NN

Enthält die ID der Pflanze, für die etwaige Antwort sein soll.

*fk\_kategorie* – INT, UN, NN

Enthält die ID der Kategorie, für die etwaige Antwort sein soll.

*antwort* – VARCHAR (45), NN

Enthält die korrekte Antwort der Kategorie und Pflanze.

**Primärschlüssel:** *fk\_pflanze*, *fk\_kategorie*

#### **Constraints:**

##### 1. *antw\_pflanze*

Fremdschlüssel: *fk\_pflanze*

Referenz: *pflanze* - id

Verhalten: On Update Cascade, On Delete Cascade

##### 2. *kat\_antw*

Fremdschlüssel: *fk\_kategorie*

Referenz: *kategorie* - id

Verhalten: On Update Cascade, On Delete Cascade

## 2.7.2: Tabelle: p\_bilder

**Engine:** InnoDB

**Default Charakter Set** = utf8;

Dient zur Lagerung der Bilder für die Pflanzen.

Table: **p\_bilder**

Columns:

<b>id</b>	int(10) UN AI PK
<b>fk_pflanze</b>	int(10) UN
<b>bild</b>	longblob

### Enthält folgende Felder:

*id* – INT, UN, NN, UQ, AI

*fk\_pflanze* – INT, UN, NN

Enthält die ID der Pflanze, für die etwaiges Bild sein soll.

*bild* – LONGBLOB

**Primärschlüssel:** *id*

### Constraints:

#### 1. *planze-bild*

Fremdschlüssel: *fk\_pflanze*

Referenz: *pflanze* - *id*

Verhalten: On Update Cascade, On Delete Cascade

### 2.7.3: Tabelle: p\_kategorien

**Engine:** InnoDB

**Default Charakter Set** = utf8;

Dient zur Lagerung der Kategorien von Pflanzen, z.B.:  
Deutscher Name = Monstera

**Table: p\_kategorien**

**Columns:**

<b>id</b>	int(10) UN AI PK
<b>kat_name</b>	varchar(45)
<b>abfrage</b>	tinyint(4)
<b>werker_gewertet</b>	tinyint(4)

**Enthält folgende Felder:**

*id* – INT, UN, NN, UQ, AI

*kat\_name* – VARCHAR (45), UQ, NN

Enthält den Namen der Kategorie.

*abfrage* - TINYINT, NN, UN, DEFAULT 1

Enthält den Wert zum Abfragen, ob diese Kategorie angezeigt/abgefragt wird.

0 = nicht sichtbar, 1 = Sichtbar

*werker\_gewertet* - TINYINT, NN, UN

Enthält den Wert zum Abfragen, ob die Kategorie für den Ausbildungstyp

Werk gewertet werden soll, oder nicht.

0 = wird nicht gewertet, 1 = wird gewertet

**Primärschlüssel:** *id*

## 2.8: Tabelle: quiz\_art

**Engine:** InnoDB

**Default Charakter Set** = utf8;

Dient zur Lagerung der Quiz-Arten/Größen.

### Enthält folgende Felder:

*id* – INT, UN, NN, UQ, AI

*name* – VARCHAR (45), UQ NN

Enthält den Namen der Quiz-Art/Größe.

*größe* – INT, UN, NN

Enthält die Anzahl der möglichen Quiz-Einträge

**Primärschlüssel:** *id*

Table: quiz\_art

### Columns:

<b>id</b>	int(10) UN AI PK
name	varchar(45)
größe	int(10) UN

## 2.8.1: Tabelle: quiz\_p\_zuweisung

**Engine:** InnoDB

**Default Charakter Set** = utf8;

Dient zur Zuweisung von Pflanzen, die der oder die Azubi/ne zu lernen hat.

<b>Table:</b> quiz_p_zuweisung	
<b>Columns:</b>	
<u>fk_azubi</u>	int(10) UN PK
<u>fk_pflanze</u>	int(10) UN PK

### Enthält folgende Felder:

*fk\_azubi* – INT, UN, NN

Enthält die ID des Azubis.

*fk\_pflanze* – INT, UN, NN,

Enthält die ID der Pflanze.

**Primärschlüssel:** *fk\_azubi*, *fk\_pflanze*

### Constraints:

#### 1. *pflanze\_quiz\_frage*

Fremdschlüssel: *fk\_pflanze*

Referenz: pflanze - id

Verhalten: On Update Cascade, On Delete Cascade

#### 2. *azubi\_quiz\_frage*

Fremdschlüssel: *fk\_azubi*

Referenz: azubis - id

Verhalten: On Update Cascade, On Delete Cascade

## 2.9: Tabelle: statistik

**Engine:** InnoDB

**Default Charakter Set** = utf8;

Dient der Lagerung von Statistiken pro abgeschlossenem Quiz.

### Enthält folgende Felder:

*id* – INT, UN, NN, UQ, AI

*log* – DATETIME NULL DEFAULT CURRENT\_TIMESTAMP

Enthält das Datum und die Zeit der Erschaffung.

*fk\_azubi* – INT, UN, NN

Enthält die ID des Azubis.

*fehlerquote* – VARCHAR(10), UN, NN,

Enthält Anzahl von Fehlern, die im Quiz gemacht wurden.

*quizzeit* – TIME, NN, DEFAULT 00:00:00

*fk\_beste\_pflanze* INT, UN, N, DEFAULT N

Enthält die ID der Pflanze mit den wenigstens Fehlern.

**Primärschlüssel:** *fk\_azubi*, *fk\_pflanze*

### Constraints:

#### 1. *stat-pflanze*

Fremdschlüssel: *fk\_pflanze*

Referenz: *pflanze* - *id*

Verhalten: On Update Cascade, On Delete Cascade

#### 2. *azubi-stat*

Fremdschlüssel: *fk\_azubi*

Referenz: *azubis* - *id*

Verhalten: On Update Cascade, On Delete Cascade

### Table: statistik

#### Columns:

<b>id</b>	int(10) UN AI PK
log	datetime
<b>fk_azubi</b>	int(10) UN
fehlerquote	varchar(10)
quizzeit	time
<b>fk_beste_pflanze</b>	int(10) UN

## 2.9.1: Tabelle: stat\_einzel

**Engine:** InnoDB

**Default Charakter Set** = utf8;

Erweiterung der Tabelle Statistik.

Dienst zur Lagerung von Statistiken pro Quiz-Frage.

Table: **stat\_einzel**

Columns:

<u>fk_statistik</u>	int(10) UN PK
<u>fk_pflanze</u>	int(10) UN PK

### Enthält folgende Felder:

*fk\_stat* – INT, UN, NN

Enthält die ID der aktuellen Quiz-Statistik

*fk\_pflanze* INT, UN, NN

Enthält die ID abgefragten Pflanze

**Primärschlüssel:** *fk\_stat*, *fk\_pflanze*

### Constraints:

#### 1. *pflanze-stat-einzel*

Fremdschlüssel: *fk\_pflanze*

Referenz: *pflanze* - id

Verhalten: On Update Cascade, On Delete Cascade

#### 2. *stat-stat\_einzel*

Fremdschlüssel: *fk\_statistik*

Referenz: *statistik* - id

Verhalten: On Update Cascade, On Delete Cascade

## 2.9.2: Tabelle: stat\_einzel\_detail

**Engine:** InnoDB

**Default Charakter Set** = utf8;

Erweiterung der Tabelle Statistik-Einzel.

Dienst zur Lagerung von eingegeben und hinterlegten  
Antworten für die Einzel-Statistik.

Table: stat_einzel_detail	
Columns:	
<u>fk_kategorie</u>	int(10) UN PK
<u>fk_statistik</u>	int(10) UN PK
<u>fk_pflanze</u>	int(10) UN PK
eingabe	varchar(45)
korrekt	varchar(45)

### Enthält folgende Felder:

*fk\_stat* – INT, UN, NN

Enthält die ID der aktuellen Quiz-Statistik, für den Fremdschlüssel.

*fk\_pflanze* INT, UN, NN

Enthält die ID abgefragten Pflanze, für den Fremdschlüssel.

*fk\_kategorie* INT, UN, NN

Enthält die ID abgefragten Kategorie

*eingabe* VARCHAR(45)

Enthält die Eingegebene Antwort

*korrekt* VARCHAR(45)

Enthält die korrekte gespeicherte Antwort

**Primärschlüssel:** *fk\_stat, fk\_pflanze, fk\_kategorie*

### Constraints:

1. *stat\_einzel-stat\_e\_details*

Fremdschlüssel: *fk\_statistik, fk\_pflanze*

Referenz: *stat\_einzel – fk\_statistik, fk\_pflanze*

Verhalten: On Update Cascade, On Delete Cascade



## 3. Die Schnittstelle:

Die aktuelle Version der Schnittstelle ist über den Endpunkt;

<http://10.33.11.142/API/dbSchnittstelle.php> zu erreichen.

Die Schnittstelle arbeitet über eine Switch-Case Funktion und selektiert anhand des übergebenen POST-Request Parameters „method=“ die genutzte Funktion.

Alle unten aufgeführten Funktionen sind in der Schnittstelle verfügbar und geben Daten im JSON-Format zurück, sofern Beispiele dazu enthalten sind.

### 3.1: Legende API:

User: der Nutzernamen

PW: das Passwort

IDaz: die ID des Azubis

IDab: die ID des Admins/Ausbilders

IDf: die ID der Fachrichtung

IDaa: die ID der Ausbildungsart

IDk: die ID der Kategorie

IDp: die ID der Pflanze

IDpb: die ID des Pflanzenbildes

IDs: die ID der Statistik

### 3.2: Login:

Die Login Funktion dient zum Login in die Lernsoftware zur Pflanzenbestimmung und der Abfrage von Nutzerdaten. – login.php

Methoden	Parameter	Optionale P.	Antwort
<i>Login</i>	<i>User, PW</i>	-	<i>JSON</i>

Beispiele:

**Anfrage:**

*http://10.33.11.142/API/dbSchnittstelle.php?*  
*method=login&User=BeispielNutzer&PW=BeispielPasswort*

**Antwort:**

Admin

```
[
  - {
    id: "1",
    nutzername: "SysAdmin",
    vorname: "Sys",
    name: "Admin",
    berflag: "1"
  },
  + {...}
]
```

Nutzer:

```
[
  - {
    id: "5",
    name: "Nutzer",
    vorname: "Beispiel",
    id_ausbilder: "1",
    id_ausbildung: "1",
    id_fachrichtung: "1",
    id_quiz_art: null,
    bool_pruefung: "1"
  }
]
```

### 3.3: Admins

Der Endpunkt "Admin" beinhaltet mehrere Funktionen. - admin.php

Methoden	Parameter	Optionale P.	Antwort
<i>getAdmins</i>	-	-	<i>JSON</i>
<i>createAdmin</i>	<i>User, PW, Vorname, Name</i>	-	-
<i>updateAdmin</i>	<i>ID, User, PW, Vorname, Name</i>	-	-

<i>deleteAdmin</i>	<i>ID</i>	-	-
--------------------	-----------	---	---

## Abfragen

Methode	Parameter	Optionale P.	Antwort
<i>getAdmins</i>	-	-	<i>JSON</i>

### Anfrage:

*http://10.33.11.142/API/dbSchnittstelle.php?*  
*method=getAdmins*

### Antwort:

```
[
  - {
    id: "1",
    nutzername: "SysAdmin",
    vorname: "Sys",
    name: "Admin",
    berflag: "1"
  },
  + {...}
]
```

## Erschaffen

Methode	Parameter	Optionale P.	Antwort
<i>createAdmin</i>	<i>User, PW, Vorname, Name</i>	-	-

---

**Anfrage:**

*http://10.33.11.142/API/dbSchnittstelle.php?*  
*method=createAdmin&User=BeispielAdmin&PW=BeispielPasswort&Vorname=BeispielVorname&Name=BeispielName*

---

## Ändern

Methode	Parameter	Optionale P.	Antwort
<i>updateAdmin</i>	<i>IDad, User, PW, Vorname, Name</i>	-	-

---

**Anfrage:**

*http://10.33.11.142/API/dbSchnittstelle.php?*  
*method=updateAdmin&IDad=Beispiel\_ID&User=BeispielAdmin&PW=BeispielPasswort&Vorname=BeispielVorname&Name=BeispielName*

---

## Löschen

Methoden	Parameter	Optionale P.	Antwort
<i>deleteAdmin</i>	<i>IDad</i>	-	-

---

**Anfrage:**

*http://10.33.11.142/API/dbSchnittstelle.php?*  
*method=deleteAdmin&IDad=Beispiel\_ID*

---

### 3.3: Azubis

Der Endpunkt "Admin" beinhaltet mehrere Funktionen. - azubis.php

Methoden	Parameter	Optionale P.	Antwort
<i>getAzubis</i>	-	-	<i>JSON</i>
<i>createAzubi</i>	<i>User, PW, IDab, IDaa, IDf, Name, Vorname</i>	-	-
<i>updateAzubi</i>	<i>ID</i>	<i>User, PW, IDab, IDaa, IDf, Name, Vorname</i>	-
<i>deleteAzubi</i>	<i>ID</i>	-	-

#### Abfragen

Methoden	Parameter	Optionale P.	Antwort
<i>getAzubis</i>	-	-	<i>JSON</i>

---

**Anfrage:**

*http://10.33.11.142/API/dbSchnittstelle.php?*  
*method=getAzubis*

**Antwort:**

```
[
  - {
      id: "1",
      name: "Voll",
      vorname: "Gala",
      nutzername: "SysUserA",
      ausbildungsart: "Vollzeit",
      ausbilder: "Sys Admin",
      fachrichtung: "Gartenlandschaftsbau",
      id_quiz_art: null,
      pruefung: "0"
    },
  + {...},
  + {...},
  + {...}
]
```

---

## Erschaffen

Methoden	Parameter	Optionale P.	Antwort
<i>createAzubi</i>	<i>User, PW, IDab, IDaa, IDf, Name, Vorname</i>	-	-

---

### Anfrage:

*http://10.33.11.142/API/dbSchnittstelle.php?*  
*method=createAzubi&User=BeispielUser&PW=BeispielPW&IDab=Beispiel\_IDab*  
*&IDaa=Beispiel\_IDaa&IDf=Beispiel\_IDf&Name=BeispielName&Vorname=BeispielVorname*

---

## Ändern

Methoden	Parameter	Optionale P.	Antwort
<i>updateAzubi</i>	<i>IDaz</i>	<i>User, PW, IDab, IDaa, IDf, Name, Vorname</i>	-

An diesem Endpunkt sind die zu ändernden Parameter zu übergeben.

---

### Anfrage:

*http://10.33.11.142/API/dbSchnittstelle.php?*  
*method=updateAdmin&IDaz=Beispiel\_IDaz*  
+ Optional:  
*&User=BeispielUser*  
*&PW=BeispielPW*  
*&IDab=Beispiel\_IDab*  
*&IDaa=Beispiel\_IDaa*  
*&IDf=Beispiel\_IDf*  
*&Name=BeispielName*  
*&Vorname=BeispielVorname*

---

## Löschen

Methoden	Parameter	Optionale P.	Antwort
<i>deleteAzubi</i>	<i>IDaz</i>	-	-

---

### Anfrage:

*http://10.33.11.142/API/dbSchnittstelle.php?*  
*method=deleteAzubi&IDaz=Beispiel\_IDaz*

---

## 3.4.1: Ausbildungsart

- ausbildungsart.php

## Abfragen

Methoden	Parameter	Optionale P.	Antwort
<i>getAusbildungsart</i>	-	-	<i>JSON</i>

---

### Anfrage:

*http://10.33.11.142/API/dbSchnittstelle.php?*  
*method=getAusbildungsart*

### Antwort:

```
[
  - {
    id: "1",
    ausbildungsart: "Vollzeit"
  },
  - {
    id: "2",
    ausbildungsart: "Werker"
  }
]
```

---

### 3.4.1: Fachrichtung

- fachrichtung.php

#### Abfragen

Methoden	Parameter	Optionale P.	Antwort
<i>getFachrichtung</i>	-	-	<i>JSON</i>

---

#### Anfrage:

*http://10.33.11.142/API/dbSchnittstelle.php?*  
*method=getFachrichtung*

#### Antwort:

```
[
  - {
    id: "1",
    fachrichtung: "Gartenlandschaftsbau"
  },
  - {
    id: "2",
    fachrichtung: "Zierpflanzenbau"
  }
]
```

---



## 3.5: Pflanzen

Der Endpunkt „Pflanzen“ beinhaltet mehrere Funktionen. – pflanzen.php

Methoden	Parameter	Optionale P.	Antwort
<i>getPflanzen</i>	-	-	<i>JSON</i>
<i>createPflanzen</i>	<i>zierbau, galabau</i>	<i>id_kategorie, antwort</i>	-
<i>deletePflanzen</i>	<i>IDp</i>	-	-

### Abfragen

Methoden	Parameter	Optionale P.	Antwort
<i>getPflanzen</i>	-	-	<i>JSON</i>

---

**Anfrage:**

*http://10.33.11.142/API/dbSchnittstelle.php?*  
*method=getPflanzen*

---

### Erschaffen

Methoden	Parameter	Optionale P.	Antwort
<i>createPflanze</i>	<i>zierbau, galabau</i>	<i>id_kategorie, antwort</i>	-

An diesem Endpunkt sperrt die API kurzzeitig die Tabelle beim Einfügen einer neuen Pflanze.

Zum erschaffen einer neuen Pflanze müssen in den ersten zwei Parametern die booleschen Werte ‚zierbau und galabau‘ übergeben werden, danach eine variable Anzahl an Parameter für die ID der Kategorie und die entsprechende Antwort.

---

**Anfrage:**

*http://10.33.11.142/API/dbSchnittstelle.php?*  
*method=createPflanze&zierbau=BeispielBool&galabau=BeispielBool&VariableK*  
*ategorie=BeispielKategorieID&VariablerAntwortName=BeispielAntwort*  
*Optional:*  
*&VariablerAntwortName=BeispielAntwort*  
*&VariablerAntwortName=BeispielAntwort*  
*&VariablerAntwortName=BeispielAntwort*

...

---

## Löschen

Methoden	Parameter	Optionale P.	Antwort
<i>deletePflanze</i>	<i>IDp</i>	-	-

---

### Anfrage:

*http://10.33.11.142/API/dbSchnittstelle.php?*  
*method=deletePflanze&IDp=Beispiel\_ID*

---

## 3.5.2 Bilder

Der Endpunkt „Pflanzen“ beinhaltet mehrere Funktionen. – pflanzen.php

Methoden	Parameter	Optionale P.	Antwort
<i>getBilder</i>	-	-	<i>JSON</i>
<i>createBild</i>	<i>Bild</i>	-	-
<i>deleteBild</i>	<i>IDpb</i>	-	-

## Abfragen

Methoden	Parameter	Optionale P.	Antwort
<i>getBilder</i>	-	-	<i>JSON</i>

---

### Anfrage:

*http://10.33.11.142/API/dbSchnittstelle.php?*  
*method=getPBilder*

### Antwort:

Eine exakte Abbildung, wäre an dieser Stelle zu groß.

```
[
  {
    "id_bild": "1",
    "bild": HIER BILD BLOB
  }
]
```

---

## Erschaffen

Methoden	Parameter	Optionale P.	Antwort
<i>createPBild</i>	-	-	-

---

**Anfrage:**

*http://10.33.11.142/API/dbSchnittstelle.php?*  
*method=***createPBild***&Bild=***BeispielBild**

---

## Löschen

Methoden	Parameter	Optionale P.	Antwort
<i>deletePBild</i>	-	-	-

---

**Anfrage:**

*http://10.33.11.142/API/dbSchnittstelle.php?*  
*method=***deletePBild***&IDpb=***Beispiel\_ID**

---

## 3.5.1: Abgefragt

Der Endpunkt „Pflanzen“ beinhaltet mehrere Funktionen. – abgefragt.php

Methoden	Parameter	Optionale P.	Antwort
<i>getAbgefragt</i>	<i>IDaz</i>	-	<i>JSON</i>
<i>createAbgefragt</i>	<i>IDaz, IDp, Counter, Gelernt</i>	-	-
<i>updateAbgefragt</i>	<i>IDaz, IDp, Counter, Gelernt</i>	-	-

## Abfragen

Methoden	Parameter	Optionale P.	Antwort
<i>getAbgefragt</i>	-	-	<i>JSON</i>

### Anfrage:

*http://10.33.11.142/API/dbSchnittstelle.php?*  
*method=getAbgefragt&IDaz=Beispiel\_ID*

### Antwort:

```
[
  - {
    IDp: "4",
    Counter: "2",
    Bool_Gelernt: "0"
  },
  - {
    IDp: "4",
    Counter: "15",
    Bool_Gelernt: "1"
  },
  - {
    IDp: "4",
    Counter: "1",
    Bool_Gelernt: "0"
  }
]
```

## Erschaffen

Methoden	Parameter	Optionale P.	Antwort
<i>createAbgefragt</i>	<i>IDaz, IDp, Counter, Gelernt(Bool)</i>	-	-

---

**Anfrage:**

*http://10.33.11.142/API/dbSchnittstelle.php?method=createAbgefragt&IDaz=Beispiel\_ID&IDp=Beispiel\_ID&Counter=Beispiel\_Counter&Gelernt=Beispiel\_Gelernt*

---

## Ändern

Methoden	Parameter	Optionale P.	Antwort
<i>updateAbgefragt</i>	<i>IDaz, IDp, Counter, Gelernt(Bool)</i>	-	-

---

**Anfrage:**

*http://10.33.11.142/API/dbSchnittstelle.php?method=updateAbgefragt&IDaz=Beispiel\_ID&IDp=Beispiel\_ID&Counter=Beispiel\_Counter&Gelernt=Beispiel\_Gelernt*

---

## 3.5.2: Kategorien

Der Endpunkt „Kategorien“ beinhaltet mehrere Funktionen. – kategorien.php

Methoden	Parameter	Optionale P.	Antwort
<i>getKategorien</i>	-	-	<i>JSON</i>
<i>createKategorien</i>	<i>Kategorie</i>	-	-
<i>updateKategorie</i>	<i>IDk, Kategorie</i>	-	-
<i>deleteKategorien</i>	<i>IDk</i>	-	-

## Abfragen

Methoden	Parameter	Optionale P.	Antwort
<i>getKategorien</i>	-	-	<i>JSON</i>

### Anfrage:

*http://10.33.11.142/API/dbSchnittstelle.php?*  
*method=getKategorien*

### Antwort:

```
[
  - {
      id: "1",
      kategorie: "Gattungsname",
      abfrage: "1"
    },
  - {
      id: "2",
      kategorie: "Artname",
      abfrage: "1"
    },
  - {
      id: "3",
      kategorie: "Denname",
      abfrage: "1"
    },
  + {...},
  + {...},
  + {...},
  + {...},
  + {...}
]
```

## Erschaffen

Methoden	Parameter	Optionale P.	Antwort
<i>createAbgefragt</i>	<i>Kategorie</i>	-	-

---

**Anfrage:**

*http://10.33.11.142/API/dbSchnittstelle.php?method=createKategorie&Kategorie=Beispiel\_Kategorie*

---

## Ändern

Methoden	Parameter	Optionale P.	Antwort
<i>updateAbgefragt</i>	<i>IDk, Kategorie</i>	-	-

---

**Anfrage:**

*http://10.33.11.142/API/dbSchnittstelle.php?method=updateKategorie&IDk=Beispiel\_ID&Kategorie=Beispiel\_Kategorie*

---

## Löschen

Methoden	Parameter	Optionale P.	Antwort
<i>deleteAbgefragt</i>	<i>IDk</i>	-	-

---

**Anfrage:**

*http://10.33.11.142/API/dbSchnittstelle.php?method=deleteKategorie&IDk=Beispiel\_ID*

---

## 3.6: Quiz

Der Endpunkt „Quiz“ beinhaltet mehrere Funktionen. – quiz.php

Methoden	Parameter	Optionale P.	Antwort
<i>getQuizArt</i>	<i>IDaz</i>	-	<i>JSON</i>
<i>createQuizArt</i>	<i>Quizname, Größe</i>	-	-
<i>updateQuizArt</i>	<i>IDqa, Quizname, Größe</i>	-	-
<i>deleteQuizArt</i>	<i>IDqa</i>	-	-
<i>getQuizPZuweisung</i>	<i>IDaz</i>	-	<i>JSON</i>
<i>createQuizPZuweisung</i>	<i>IDaz, IDp</i>	-	
<i>deleteQuizPZuweisung</i>	<i>IDaz, IDp</i>	-	

## Abfragen

Methoden	Parameter	Optionale P.	Antwort
<i>getQuizArt</i>	<i>IDaz</i>	-	<i>JSON</i>

---

**Anfrage:**

*http://10.33.11.142/API/dbSchnittstelle.php?*  
*method=getQuizArt*

**Antwort:**

```
[  
  - {  
    id: "1",  
    quizname: "BeispielQuizGroß",  
    quizgröße: "100"  
  },  
  - {  
    id: "2",  
    quizname: "BeispielQuizKlein",  
    quizgröße: "25"  
  }  
]
```



Methoden	Parameter	Optionale P.	Antwort
<i>getQuizPZuweisung</i>	<i>IDaz</i>	-	<i>JSON</i>

---

**Anfrage:**

*http://10.33.11.142/API/dbSchnittstelle.php?*  
*method=getQuizPZuweisung*

**Antwort:**

```
[
  - {
    id_pflanze: "1"
  },
  - {
    id_pflanze: "2"
  },
  - {
    id_pflanze: "3"
  }
]
```

---

## Erschaffen

Methoden	Parameter	Optionale P.	Antwort
<i>createQuizArt</i>	<i>Quizname, Größe</i>	-	-

---

**Anfrage:**

*http://10.33.11.142/API/dbSchnittstelle.php?method=createQuizArt&Quizname*  
*=Beispiel\_Name&Größe=Beispiel\_Größe*

---

Methoden	Parameter	Optionale P.	Antwort
<i>createQuizPZuweisung</i>	<i>IDaz, IDp</i>	-	

---

**Anfrage:**

*http://10.33.11.142/API/dbSchnittstelle.php?method=createQuizpZuweisung&IDaz=Beispiel\_ID&IDp=Beispiel\_ID*

---

## Ändern

Methoden	Parameter	Optionale P.	Antwort
<i>updateQuizArt</i>	<i>IDqa, Quizname, Größe</i>	-	-

---

**Anfrage:**

*http://10.33.11.142/API/dbSchnittstelle.php?method=updateQuizAr&IDqa=Beispiel\_ID&Quizname=Beispiel\_Name&Größe=Beispiel\_Größe*

---

## Löschen

Methoden	Parameter	Optionale P.	Antwort
<i>deleteQuizArt</i>	<i>IDqa</i>	-	-

---

**Anfrage:**

*http://10.33.11.142/API/dbSchnittstelle.php?*  
*method=deleteQuizArt&IDqa=Beispiel\_ID*

---

Methoden	Parameter	Optionale P.	Antwort
<i>deleteQuizPZuweisung</i>	<i>IDaz, IDp</i>	-	

---

**Anfrage:**

*http://10.33.11.142/API/dbSchnittstelle.php?*  
*method=deleteQuizPZuweisung&IDaz=Beispiel\_ID&IDp=Beispiel\_ID*

---

### 3.6.1: Statistik

Der Endpunkt „Statistik“ beinhaltet mehrere Funktionen. – statistik.php

Methoden	Parameter	Optionale P.	Antwort
<i>getStatList</i>	<i>IDaz</i>	-	JSON
<i>getStatistik</i>	<i>IDs</i>	-	JSON
<i>createStatistik</i>	<i>IDaz</i> , <i>FQuote</i> , <i>Zeit</i> , <i>IDp</i>	-	-
<i>createStatEinzel</i>	<i>IDs</i> , <i>IDk</i> , <i>IDp</i> , <i>Eingabe</i>		
<i>updateStatistik</i>	<i>IDaz</i> , <i>FQuote</i> , <i>Zeit</i> , <i>IDp</i>	-	-

### Abfragen

Methoden	Parameter	Optionale P.	Antwort
<i>getStatList</i>	<i>IDaz</i>	-	JSON

**Anfrage:**

*http://10.33.11.142/API/dbSchnittstelle.php?*  
*method=getStatList&IDaz=Beispiel\_ID*

**Antwort:**

```
[
  - {
    id_statistik: "1",
    erstellt: "2020-10-01 13:49:58",
    fehlerquote: "0",
    zeit: "00:02:30",
    id_beste_pflanze: "1"
  },
  - {
    id_statistik: "2",
    erstellt: "2020-10-01 13:49:58",
    fehlerquote: "1",
    zeit: "00:01:30",
    id_beste_pflanze: "1"
  },
  + { ... },
  + { ... },
  + { ... },
  + { ... },
  + { ... }
]
```

Methoden	Parameter	Optionale P.	Antwort
<i>getStatistik</i>	<i>IDs</i>	-	<i>JSON</i>

### Anfrage:

*http://10.33.11.142/API/dbSchnittstelle.php?*  
*method=getStatistik&IDs=Beispiel\_ID*

### Antwort:

```
[
  - {
    id_statistik: "1",
    erstellt: "2020-10-01 13:49:58",
    fehlerquote: "0",
    zeit: "00:02:30",
    id_beste_pflanze: "1",
    - pflanzen: [
      - {
        id_pflanze: "1",
        - antworten: [
          - {
            kategorie: "Gattungsname",
            eingabe: "Gattung A",
            korrekt: "Gattungsname A"
          },
          - {
            kategorie: "Artname",
            eingabe: "Artname A",
            korrekt: "Artname A"
          }
        ]
      },
      - {
        id_pflanze: "2",
        - antworten: [
          - {
            kategorie: "Gattungsname",
            eingabe: "Gattung B",
            korrekt: "Gattungsname B"
          },
          - {
            kategorie: "Artname",
            eingabe: "MIAU B",
            korrekt: "Artname B"
          }
        ]
      }
    ]
  }
]
```

## Erschaffen

Methoden	Parameter	Optionale P.	Antwort
<i>createStatistik</i>	<i>IDaz, FQuote, Zeit, IDp</i>	-	-

---

**Anfrage:**

*http://10.33.11.142/API/dbSchnittstelle.php?method=createStatistik&IDaz=Beispiel\_ID&FQuote=Beispiel\_Quote&Zeit=Beispiel\_Zeit&IDp=Beispiel\_ID*

---

Methoden	Parameter	Optionale P.	Antwort
<i>createStatEinzel</i>	<i>IDs, IDk, IDp, Eingabe</i>		

---

**Anfrage:**

*http://10.33.11.142/API/dbSchnittstelle.php?method=createStatEinzel&IDs=Beispiel\_ID&IDk=Beispiel\_ID&IDp=Beispiel\_ID&Eingabe=Beispiel\_Eingabe*

## Ändern

Methoden	Parameter	Optionale P.	Antwort
<i>updateStatistik</i>	<i>IDaz, FQuote, Zeit, IDp</i>	-	-

---

**Anfrage:**

*http://10.33.11.142/API/dbSchnittstelle.php?method=updateStatistik&IDaz=Beispiel\_ID&FQuote=Beispiel\_Quote&Zeit=Beispiel\_Zeit&IDp=Beispiel\_ID*

---