

Inhaltsverzeichnis

Update

- [updateabgeleitetbydkey.md](#)
- [updateabgeleitetbytkey.md](#)
- [updatearbeitetbykkey.md](#)
- [updatearbeitetbymkey.md](#)
- [updateartmain.md](#)
- [updatebearbeitetbymkey.md](#)
- [updatebearbeitetbytkey.md](#)
- [updatebezahlungsartmain.md](#)
- [updatedienstleistungmain.md](#)
- [updatekundemain.md](#)
- [updatemitarbeitermain.md](#)
- [updateortmain.md](#)
- [updateprioritaetmain.md](#)
- [updaterechnungmain.md](#)
- [updaterechtegruppemain.md](#)
- [updateresidiertbykkey.md](#)
- [updateresidiertbyokey.md](#)
- [updatestatusmain.md](#)
- [updateticketmain.md](#)

Delete

- [deleteabgeleitetbydkey.md](#)
- [deleteabgeleitetbytkey.md](#)
- [deletearbeitetbykkey.md](#)
- [deletearbeitetbymkey.md](#)
- [deleteartmain.md](#)
- [deletebearbeitetbymkey.md](#)
- [deletebearbeitetbytkey.md](#)
- [deletebezahlungsartmain.md](#)
- [deletedienstleistungmain.md](#)
- [deletekundemain.md](#)
- [deletemitarbeitermain.md](#)
- [deleteortmain.md](#)
- [deleteprioritaetmain.md](#)
- [deleterechnungmain.md](#)
- [deleterechtegruppemain.md](#)
- [deleteresidiertbykkey.md](#)
- [deleteresidiertbyokey.md](#)
- [deletestatusmain.md](#)
- [deleteticketmain.md](#)

Set

- [setartmain.md](#)
- [setbezahlungsartmain.md](#)
- [setdienstleistungmain.md](#)
- [setkundemain.md](#)
- [setmitarbeitermain.md](#)
- [setortmain.md](#)
- [setprioritaetmain.md](#)
- [setrechnungmain.md](#)
- [setrechtgruppemain.md](#)
- [setstatusmain.md](#)
- [setticketmain.md](#)

Get

- [getabgeleitetbykey.md](#)
- [getarbeitetbymkey.md](#)
- [getartall.md](#)
- [getartbykey.md](#)
- [getbearbeitetbykey.md](#)
- [getbezahlungsartall.md](#)
- [getbezahlungsartbykey.md](#)
- [getdienstleistungall.md](#)
- [getdienstleistungbykey.md](#)
- [getkundeall.md](#)
- [getkundebykey.md](#)
- [getmitarbeiterall.md](#)
- [getmitarbeiterbykey.md](#)
- [getortall.md](#)
- [getortbykey.md](#)
- [getprioritaetall.md](#)
- [getprioritaetbykey.md](#)
- [getrechnungall.md](#)
- [getrechnungbykey.md](#)
- [getrechnungbykey.md](#)
- [getrechnungbykey.md](#)
- [getrechtgruppeall.md](#)
- [getrechtgruppebykey.md](#)
- [getresidiertbykey.md](#)
- [getstatusall.md](#)
- [getstatusbykey.md](#)
- [getticketall.md](#)
- [getticketbykey.md](#)

updateabgeleitetbydkey.md

[Zurück zum Inhaltsverzeichnis](#)



| Name | Type |
|---|---------|
| FK_Dienstleistung  | int(11) |
| FK_Ticket  | int(11) |

Figure 1: Database Image of Table abgeleitet

Method: `updateabgeleitetbydkey`

Der Endpoint `updateabgeleitetbydkey` kann in `abgeleitet` aktualisieren.

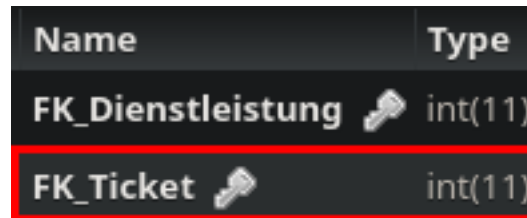
Dieser Endpunkt muss mit folgenden Parametern aufgerufen werden:

`newFK_Ticket` mit Type `integer`

`key`(Rot markiert im Bild) mit Type `integer`

updateabgeleitetbytkey.md

[Zurück zum Inhaltsverzeichnis](#)





| Name | Type |
|---|---------|
| FK_Dienstleistung  | int(11) |
| FK_Ticket  | int(11) |

Figure 2: Database Image of Table abgeleitet

Method: `updateabgeleitetbytkey`

Der Endpoint `updateabgeleitetbytkey` kann in `abgeleitet` aktualisieren.

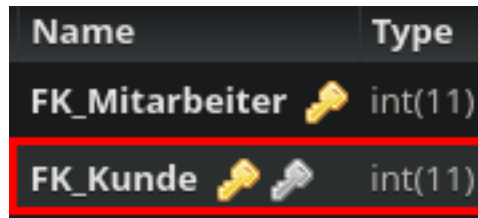
Dieser Endpunkt muss mit folgenden Parametern aufgerufen werden:

`newFK_Dienstleistung` mit Type `integer`

`key`(Rot markiert im Bild) mit Type `integer`

updatearbeitbykey.md

[Zurück zum Inhaltsverzeichnis](#)






| Name | Type |
|--|---------|
| FK_Mitarbeiter  | int(11) |
| FK_Kunde   | int(11) |

Figure 3: Database Image of Table arbeit

Method: `updatearbeitbykey`

Der Endpoint `updatearbeitbykey` kann in `arbeit` aktualisieren.

Dieser Endpunkt muss mit folgenden Parametern aufgerufen werden:

`newFK_Mitarbeiter` mit Type `integer`

`key`(Rot markiert im Bild) mit Type `integer`

updatearbeitetbymkey.md

[Zurück zum Inhaltsverzeichnis](#)




| Name | Type |
|--|---------|
| FK_Mitarbeiter  | int(11) |
| FK_Kunde   | int(11) |

Figure 4: Database Image of Table arbeitet

Method: `updatearbeitetbymkey`

Der Endpoint `updatearbeitetbymkey` kann in `arbeitet` aktualisieren.

Dieser Endpunkt muss mit folgenden Parametern aufgerufen werden:

`newFK_Kunde` mit Type `integer`

`key`(Rot markiert im Bild) mit Type `integer`

updateartmain.md

[Zurück zum Inhaltsverzeichnis](#)


| Name | Type |
|--|--------------|
| PK_Art  | int(11) |
| Artname | varchar(255) |
| Beschreibung | text |

Figure 5: Database Image of Table art

Method: `updateartmain`

Der Endpoint `updateartmain` kann in `art` aktualisieren.

Dieser Endpoint muss mit folgenden Parametern aufgerufen werden:

`Beschreibung` mit Type `string`

`Artname` mit Type `string`

`key`(Rot markiert im Bild) mit Type `integer`

updatebearbeitetbymkey.md

[Zurück zum Inhaltsverzeichnis](#)






| Name | Type |
|---|---------|
| FK_Mitarbeiter  | int(11) |
| FK_Ticket   | int(11) |

Figure 6: Database Image of Table `bearbeitet`

Method: `updatebearbeitetbymkey`

Der Endpoint `updatebearbeitetbymkey` kann in `bearbeitet` aktualisieren.

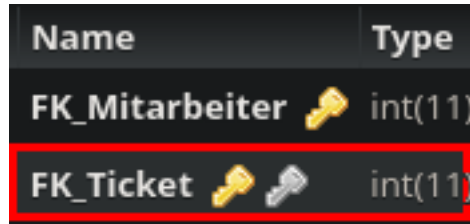
Dieser Endpoint muss mit folgenden Parametern aufgerufen werden:

`new_FK_Ticket` mit Type `integer`

`FK_Mitarbeiter` mit Type `integer`

updatebearbeitetbytkey.md

[Zurück zum Inhaltsverzeichnis](#)






| Name | Type |
|---|---------|
| FK_Mitarbeiter  | int(11) |
| FK_Ticket   | int(11) |

Figure 7: Database Image of Table `bearbeitet`

Method: `updatebearbeitetbytkey`

Der Endpoint `updatebearbeitetbytkey` kann in `bearbeitet` aktualisieren.

Dieser Endpoint muss mit folgenden Parametern aufgerufen werden:

`new_FK_Mitarbeiter` mit Type `integer`

`FK_Ticket` mit Type `integer`

updatebezahlungsartmain.md

[Zurück zum Inhaltsverzeichnis](#)


| Name | Type |
|---|--------------|
| Artname | varchar(255) |
| Beschreibung | text |
| PK_Bezahlungsart  | int(11) |

Figure 8: Database Image of Table bezahlungsart

Method: updatebezahlungsartmain

Der Endpoint updatebezahlungsartmain kann in bezahlungsart aktualisieren.

Dieser Endpoint muss mit folgenden Parametern aufgerufen werden:

Beschreibung mit Type string

Artname mit Type string

PK_Bezahlungsart mit Type integer

updatedienstleistungmain.md

[Zurück zum Inhaltsverzeichnis](#)


| Name | Type |
|--|---------------|
| PK_Dienstleistung  | int(11) |
| Beschreibung | text |
| Kosten | decimal(10,2) |

Figure 9: Database Image of Table dienstleistung

Method: updatedienstleistungmain

Der Endpoint updatedienstleistungmain kann in dienstleistung aktualisieren.

Dieser Endpunkt muss mit folgenden Parametern aufgerufen werden:

Kosten mit Type double

Beschreibung mit Type string

PK_Dienstleistung mit Type integer

updatekundemain.md

[Zurück zum Inhaltsverzeichnis](#)


| Name | Type |
|---|--------------|
| PK_Kunde  | int(11) |
| Kundennummer | varchar(255) |
| Firmenname | varchar(255) |
| Email | varchar(255) |
| Passwort | varchar(255) |

Figure 10: Database Image of Table kunde

Method: updatekundemain

Der Endpoint updatekundemain kann in kunde aktualisieren.

Dieser Endpoint muss mit folgenden Parametern aufgerufen werden:

Kundennummer mit Type string

Firmenname mit Type string

Email mit Type string

Passwort mit Type string

PK_Kunde mit Type integer

updatemitarbeitermain.md

[Zurück zum Inhaltsverzeichnis](#)

| Name | Type |
|--|--------------|
| PK_Mitarbeiter  | int(11) |
| Vorname | varchar(255) |
| Nachname | varchar(255) |
| Position | varchar(255) |
| Festnetznummer | varchar(20) |
| Mobilnummer | varchar(20) |
| Email | varchar(255) |
| Passwort | varchar(255) |
| FK_Ort  | int(11) |
| FK_Rechtegruppe  | int(11) |

Figure 11: Database Image of Table mitarbeiter

Method: updatemitarbeitermain

Der Endpoint updatemitarbeitermain kann in mitarbeiter aktualisieren.

Dieser Endpunkt muss mit folgenden Parametern aufgerufen werden:

Vorname mit Type string

Nachname mit Type string

Position mit Type string

Festnetznummer mit Type string

Mobilnummer mit Type string

Email mit Type string

Passwort mit Type `string`
FK_Ort mit Type `integer`
FK_Rechtegruppe mit Type `integer`
PK_Mitarbeiter mit Type `integer`
FK_Kunde mit Type `integer`

updateortmain.md

[Zurück zum Inhaltsverzeichnis](#)


| Name | Type |
|---|--------------|
| PK_Ort  | int(11) |
| Stadt | varchar(255) |
| PLZ | varchar(10) |
| Straße | varchar(255) |
| Hausnummer | varchar(10) |

Figure 12: Database Image of Table ort

Method: `updateortmain`

Der Endpoint `updateortmain` kann in `ort` aktualisieren.

Dieser Endpunkt muss mit folgenden Parametern aufgerufen werden:

`Stadt` mit Type `string`

`PLZ` mit Type `string`

`Straße` mit Type `string`

`Hausnummer` mit Type `string`

`PK_Ort` mit Type `integer`

updateprioritaetmain.md

[Zurück zum Inhaltsverzeichnis](#)

| Name | Type |
|---|--------------|
| PK_Priorität  | int(11) |
| Prioritätsname | varchar(255) |
| Beschreibung | text |

Figure 13: Database Image of Table prioritael

Method: updateprioritaetmain

Der Endpoint updateprioritaetmain kann in prioritael aktualisieren.

Dieser Endpunkt muss mit folgenden Parametern aufgerufen werden:

Prioritätsname mit Type string

Beschreibung mit Type string

PK_Priorität mit Type integer

updaterechnungmain.md

[Zurück zum Inhaltsverzeichnis](#)





| Name | Type |
|---|--------------|
| Rechnungsnr  | int(11) |
| Ticketbeschreibung | text |
| Abrechnung | varchar(255) |
| Ratenzahlung | tinyint(1) |
| FK_Bezahlungsart  | int(11) |
| FK_Kunde  | int(11) |
| FK_Ticket  | int(11) |

Figure 14: Database Image of Table rechnung

Method: updaterechnungmain

Der Endpoint updaterechnungmain kann in rechnung aktualisieren.

Dieser Endpoint muss mit folgenden Parametern aufgerufen werden:

Ticketbeschreibung mit Type string

Ratenzahlung mit Type integer

FK_Bezahlungsart mit Type integer

Rechnungsnr mit Type integer

updaterechtegruppemain.md

[Zurück zum Inhaltsverzeichnis](#)


| Name | Type |
|--|---------------|
| Administrationsrechte | tinyint(1) |
| Bestelllimit | decimal(10,2) |
| Helpdesk_Fernwartung | tinyint(1) |
| PK_Rechtegruppe  | int(11) |

Figure 15: Database Image of Table rechtegruppe

Method: updaterechtegruppemain

Der Endpoint updaterechtegruppemain kann in rechtegruppe aktualisieren.

Dieser Endpunkt muss mit folgenden Parametern aufgerufen werden:

Administrationsrechte mit Type integer

Bestelllimit mit Type double

Helpdesk_Fernwartung mit Type integer

PK_Rechtegruppe mit Type integer

updateresidiertbykkey.md

[Zurück zum Inhaltsverzeichnis](#)

| Name | Type |
|--|----------|
| FK_Kunde   | int(11) |
| FK_Ort   | int(255) |

Figure 16: Database Image of Table `residiert`

Method: `updateresidiertbykkey`

Der Endpoint `updateresidiertbykkey` kann in `residiert` aktualisieren.

Dieser Endpunkt muss mit folgenden Parametern aufgerufen werden:

`FK_Ort` mit Type `integer`

`FK_Kunde` mit Type `integer`

updateresidiertbykey.md

[Zurück zum Inhaltsverzeichnis](#)




| Name | Type |
|--|----------|
| FK_Kunde   | int(11) |
| FK_Ort   | int(255) |

Figure 17: Database Image of Table `residiert`

Method: `updateresidiertbykey`

Der Endpoint `updateresidiertbykey` kann in `residiert` aktualisieren.

Dieser Endpunkt muss mit folgenden Parametern aufgerufen werden:

FK_Kunde mit Type `integer`

FK_Ort mit Type `integer`

updatestatusmain.md

[Zurück zum Inhaltsverzeichnis](#)


| Name | Type |
|--|--------------|
| PK_Status  | int(11) |
| Statusname | varchar(255) |
| Farbe | varchar(20) |
| Beschreibung | text |

Figure 18: Database Image of Table status

Method: `updatestatusmain`

Der Endpoint `updatestatusmain` kann in `status` aktualisieren.

Dieser Endpunkt muss mit folgenden Parametern aufgerufen werden:

`Statusname` mit Type `string`

`Farbe` mit Type `string`

`Beschreibung` mit Type `string`

`PK_Status` mit Type `integer`

updateticketmain.md

[Zurück zum Inhaltsverzeichnis](#)







| Name | Type |
|---|-------------|
| PK_Ticket  | int(11) |
| Ticket_Titel | varchar(75) |
| Beschreibung | text |
| InterneNotiz | text |
| InternerStatus | text |
| Erstelldatum | datetime |
| Bearbeitungsdatum | datetime |
| FK_Status  | int(11) |
| FK_Priorität  | int(11) |
| FK_Art  | int(11) |
| FK_Mitarbeiter  | int(11) |
| FK_Kunde  | int(11) |

Figure 19: Database Image of Table ticket

Method: updateticketmain

Der Endpoint updateticketmain kann in ticket aktualisieren.

Dieser Endpoint muss mit folgenden Parametern aufgerufen werden:

Ticket_Titel mit Type string

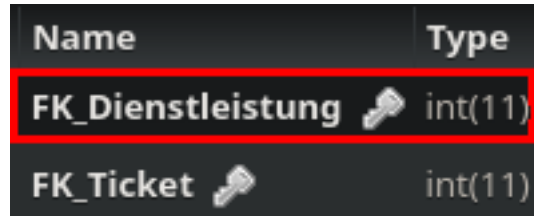
Beschreibung mit Type string

InterneNotiz mit Type string

InternerStatus mit Type string
Erstelldatum mit Type string
Bearbeitungsdatum mit Type string
FK_Status mit Type integer
FK_Prioritaet mit Type integer
FK_Art mit Type integer
FK_Mitarbeiter mit Type integer
FK_Kunde mit Type integer
PK_Ticket mit Type integer
Dienstleistungen mit Type array

deleteabgeleitetbydkey.md

[Zurück zum Inhaltsverzeichnis](#)





| Name | Type |
|---|---------|
| FK_Dienstleistung  | int(11) |
| FK_Ticket  | int(11) |

Figure 20: Database Image of Table abgeleitet

Method: `deleteabgeleitetbydkey`

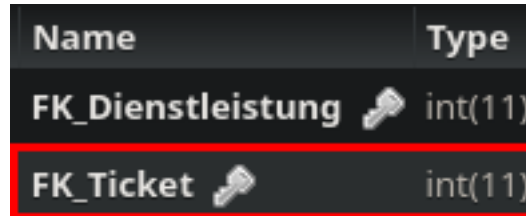
Der Endpoint `deleteabgeleitetbydkey` kann in `abgeleitet` löschen.

Dieser Endpunkt muss mit folgenden Parametern aufgerufen werden:

`key`(Rot markiert im Bild) mit Type `integer`

deleteabgeleitetbytkey.md

[Zurück zum Inhaltsverzeichnis](#)





| Name | Type |
|---|---------|
| FK_Dienstleistung  | int(11) |
| FK_Ticket  | int(11) |

Figure 21: Database Image of Table `abgeleitet`

Method: `deleteabgeleitetbytkey`

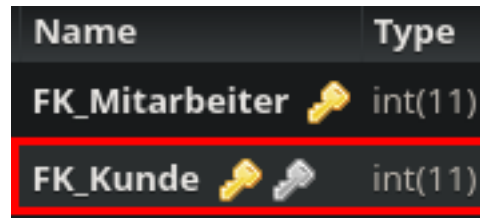
Der Endpoint `deleteabgeleitetbytkey` kann in `abgeleitet` löschen.

Dieser Endpunkt muss mit folgenden Parametern aufgerufen werden:

`key`(Rot markiert im Bild) mit Type `integer`

deletearbeitetbykey.md

[Zurück zum Inhaltsverzeichnis](#)





| Name | Type |
|--|---------|
| FK_Mitarbeiter  | int(11) |
| FK_Kunde   | int(11) |

Figure 22: Database Image of Table arbeitet

Method: `deletearbeitetbykey`

Der Endpoint `deletearbeitetbykey` kann in `arbeitet` löschen.

Dieser Endpunkt muss mit folgenden Parametern aufgerufen werden:

`key`(Rot markiert im Bild) mit Type `integer`

deletearbeitetbymkey.md

[Zurück zum Inhaltsverzeichnis](#)



| Name | Type |
|--|---------|
| FK_Mitarbeiter  | int(11) |
| FK_Kunde   | int(11) |

Figure 23: Database Image of Table arbeitet

Method: `deletearbeitetbymkey`

Der Endpoint `deletearbeitetbymkey` kann in `arbeitet` löschen.

Dieser Endpunkt muss mit folgenden Parametern aufgerufen werden:

`key`(Rot markiert im Bild) mit Type `integer`

deleteartmain.md

[Zurück zum Inhaltsverzeichnis](#)


| Name | Type |
|--|--------------|
| PK_Art  | int(11) |
| Artname | varchar(255) |
| Beschreibung | text |

Figure 24: Database Image of Table art

Method: `deleteartmain`

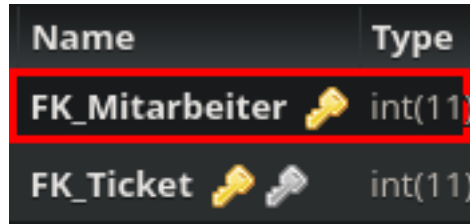
Der Endpoint `deleteartmain` kann in `art` löschen.

Dieser Endpoint muss mit folgenden Parametern aufgerufen werden:

`key`(Rot markiert im Bild) mit Type `integer`

deletebearbeitetbymkey.md

[Zurück zum Inhaltsverzeichnis](#)



| Name | Type |
|----------------|---------|
| FK_Mitarbeiter | int(11) |
| FK_Ticket | int(11) |

Figure 25: Database Image of Table `bearbeitet`

Method: `deletebearbeitetbymkey`

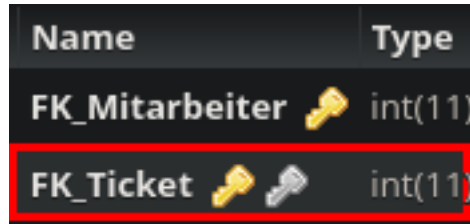
Der Endpoint `deletebearbeitetbymkey` kann in `bearbeitet` löschen.

Dieser Endpoint muss mit folgenden Parametern aufgerufen werden:

`key`(Rot markiert im Bild) mit Type `integer`

deletebearbeitetbytkey.md

[Zurück zum Inhaltsverzeichnis](#)



| Name | Type |
|----------------|---------|
| FK_Mitarbeiter | int(11) |
| FK_Ticket | int(11) |

Figure 26: Database Image of Table `bearbeitet`

Method: `deletebearbeitetbytkey`

Der Endpoint `deletebearbeitetbytkey` kann in `bearbeitet` löschen.

Dieser Endpoint muss mit folgenden Parametern aufgerufen werden:

`key` (Rot markiert im Bild) mit Type `integer`

deletebezahlungsartmain.md

[Zurück zum Inhaltsverzeichnis](#)


| Name | Type |
|---|--------------|
| Artname | varchar(255) |
| Beschreibung | text |
| PK_Bezahlungsart  | int(11) |

Figure 27: Database Image of Table bezahlungsart

Method: `deletebezahlungsartmain`

Der Endpoint `deletebezahlungsartmain` kann in `bezahlungsart` löschen.

Dieser Endpoint muss mit folgenden Parametern aufgerufen werden:

`key`(Rot markiert im Bild) mit Type `integer`

deletedienstleistungmain.md

[Zurück zum Inhaltsverzeichnis](#)


| Name | Type |
|--|---------------|
| PK_Dienstleistung  | int(11) |
| Beschreibung | text |
| Kosten | decimal(10,2) |

Figure 28: Database Image of Table dienstleistung

Method: `deletedienstleistungmain`

Der Endpoint `deletedienstleistungmain` kann in `dienstleistung` löschen.

Dieser Endpunkt muss mit folgenden Parametern aufgerufen werden:

`key`(Rot markiert im Bild) mit Type `integer`

deletetekundemain.md

[Zurück zum Inhaltsverzeichnis](#)


| Name | Type |
|---|--------------|
| PK_Kunde  | int(11) |
| Kundennummer | varchar(255) |
| Firmenname | varchar(255) |
| Email | varchar(255) |
| Passwort | varchar(255) |

Figure 29: Database Image of Table kunde

Method: `deletetekundemain`

Der Endpoint `deletetekundemain` kann in `kunde` löschen.

Dieser Endpunkt muss mit folgenden Parametern aufgerufen werden:

`key`(Rot markiert im Bild) mit Type `integer`

deletemitarbeitermain.md

[Zurück zum Inhaltsverzeichnis](#)

| Name | Type |
|--|--------------|
| PK_Mitarbeiter  | int(11) |
| Vorname | varchar(255) |
| Nachname | varchar(255) |
| Position | varchar(255) |
| Festnetznummer | varchar(20) |
| Mobilnummer | varchar(20) |
| Email | varchar(255) |
| Passwort | varchar(255) |
| FK_Ort  | int(11) |
| FK_Rechtegruppe  | int(11) |

Figure 30: Database Image of Table mitarbeiter

Method: `deletemitarbeitermain`

Der Endpoint `deletemitarbeitermain` kann in `mitarbeiter` löschen.

Dieser Endpunkt muss mit folgenden Parametern aufgerufen werden:

`key`(Rot markiert im Bild) mit Type `integer`

deleteortmain.md

[Zurück zum Inhaltsverzeichnis](#)

| Name | Type |
|---|--------------|
| PK_Ort  | int(11) |
| Stadt | varchar(255) |
| PLZ | varchar(10) |
| Straße | varchar(255) |
| Hausnummer | varchar(10) |

Figure 31: Database Image of Table ort

Method: `deleteortmain`

Der Endpoint `deleteortmain` kann in `ort` löschen.

Dieser Endpunkt muss mit folgenden Parametern aufgerufen werden:

`key`(Rot markiert im Bild) mit Type `integer`

deleteprioritaetmain.md

[Zurück zum Inhaltsverzeichnis](#)

| Name | Type |
|-----------------------|--------------|
| PK_Priorität 🔑 | int(11) |
| Prioritätsname | varchar(255) |
| Beschreibung | text |

Figure 32: Database Image of Table prioritaet

Method: `deleteprioritaetmain`

Der Endpoint `deleteprioritaetmain` kann in `prioritaet` löschen.

Dieser Endpoint muss mit folgenden Parametern aufgerufen werden:

`key` (Rot markiert im Bild) mit Type `integer`

deleterrechnungmain.md

[Zurück zum Inhaltsverzeichnis](#)





| Name | Type |
|--|--------------|
| Rechnungsnr  | int(11) |
| Ticketbeschreibung | text |
| Abrechnung | varchar(255) |
| Ratenzahlung | tinyint(1) |
| FK_Bezahlungsart  | int(11) |
| FK_Kunde  | int(11) |
| FK_Ticket  | int(11) |

Figure 33: Database Image of Table rechnung

Method: `deleterrechnungmain`

Der Endpoint `deleterrechnungmain` kann in `rechnung` löschen.

Dieser Endpoint muss mit folgenden Parametern aufgerufen werden:

`key`(Rot markiert im Bild) mit Type `integer`

deleterechtegruppemain.md

[Zurück zum Inhaltsverzeichnis](#)


| Name | Type |
|--|---------------|
| Administrationsrechte | tinyint(1) |
| Bestelllimit | decimal(10,2) |
| Helpdesk_Fernwartung | tinyint(1) |
| PK_Rechtegruppe  | int(11) |

Figure 34: Database Image of Table rechtegruppe

Method: `deleterechtegruppemain`

Der Endpoint `deleterechtegruppemain` kann in `rechtegruppe` löschen.

Dieser Endpunkt muss mit folgenden Parametern aufgerufen werden:

`key`(Rot markiert im Bild) mit Type `integer`

deleteresidiertbykkey.md

[Zurück zum Inhaltsverzeichnis](#)

| Name | Type |
|--|----------|
| FK_Kunde   | int(11) |
| FK_Ort   | int(255) |

Figure 35: Database Image of Table residiert

Method: `deleteresidiertbykkey`

Der Endpoint `deleteresidiertbykkey` kann in `residiert` löschen.

Dieser Endpunkt muss mit folgenden Parametern aufgerufen werden:

`key` (Rot markiert im Bild) mit Type `integer`

deleteresidiertbykey.md

[Zurück zum Inhaltsverzeichnis](#)




| Name | Type |
|--|----------|
| FK_Kunde   | int(11) |
| FK_Ort   | int(255) |

Figure 36: Database Image of Table residiert

Method: deleteresidiertbykey

Der Endpoint deleteresidiertbykey kann in residiert löschen.

Dieser Endpunkt muss mit folgenden Parametern aufgerufen werden:

key(Rot markiert im Bild) mit Type integer

deletestatusmain.md

[Zurück zum Inhaltsverzeichnis](#)


| Name | Type |
|--|--------------|
| PK_Status  | int(11) |
| Statusname | varchar(255) |
| Farbe | varchar(20) |
| Beschreibung | text |

Figure 37: Database Image of Table status

Method: `deletestatusmain`

Der Endpoint `deletestatusmain` kann in `status` löschen.

Dieser Endpunkt muss mit folgenden Parametern aufgerufen werden:

`key`(Rot markiert im Bild) mit Type `integer`

deleteticketmain.md

[Zurück zum Inhaltsverzeichnis](#)




| Name | Type |
|---|-------------|
| PK_Ticket  | int(11) |
| Ticket_Titel | varchar(75) |
| Beschreibung | text |
| InterneNotiz | text |
| InternerStatus | text |
| Erstelldatum | datetime |
| Bearbeitungsdatum | datetime |
| FK_Status  | int(11) |
| FK_Priorität  | int(11) |
| FK_Art  | int(11) |
| FK_Mitarbeiter  | int(11) |
| FK_Kunde  | int(11) |

Figure 38: Database Image of Table ticket

Method: deleteticketmain

Der Endpoint deleteticketmain kann in ticket löschen.

Dieser Endpunkt muss mit folgenden Parametern aufgerufen werden:

key(Rot markiert im Bild) mit Type integer

setartmain.md

[Zurück zum Inhaltsverzeichnis](#)


| Name | Type |
|--|--------------|
| PK_Art  | int(11) |
| Artname | varchar(255) |
| Beschreibung | text |

Figure 39: Database Image of Table art

Method: `setartmain`

Der Endpoint `setartmain` kann in `art` erstellen.

Dieser Endpoint muss mit folgenden Parametern aufgerufen werden:

`artname` mit Type `string`

`beschreibung` mit Type `string`

setbezahlungsartmain.md

[Zurück zum Inhaltsverzeichnis](#)


| Name | Type |
|--|--------------|
| Artname | varchar(255) |
| Beschreibung | text |
| PK_Bezahlungsart  | int(11) |

Figure 40: Database Image of Table bezahlungsart

Method: `setbezahlungsartmain`

Der Endpoint `setbezahlungsartmain` kann in `bezahlungsart` erstellen.

Dieser Endpoint muss mit folgenden Parametern aufgerufen werden:

`artname` mit Type `string`

`beschreibung` mit Type `string`

setdienstleistungmain.md

[Zurück zum Inhaltsverzeichnis](#)


| Name | Type |
|---|---------------|
| PK_Dienstleistung  | int(11) |
| Beschreibung | text |
| Kosten | decimal(10,2) |

Figure 41: Database Image of Table dienstleistung

Method: `setdienstleistungmain`

Der Endpoint `setdienstleistungmain` kann in `dienstleistung` erstellen.

Dieser Endpoint muss mit folgenden Parametern aufgerufen werden:

`beschreibung` mit Type `string`

`kosten` mit Type `double`

setkundemain.md

[Zurück zum Inhaltsverzeichnis](#)


| Name | Type |
|--|--------------|
| PK_Kunde  | int(11) |
| Kundennummer | varchar(255) |
| Firmenname | varchar(255) |
| Email | varchar(255) |
| Passwort | varchar(255) |

Figure 42: Database Image of Table kunde

Method: `setkundemain`

Der Endpoint `setkundemain` kann in `kunde` erstellen.

Dieser Endpoint muss mit folgenden Parametern aufgerufen werden:

Kundennummer mit Type `string`

Firmenname mit Type `string`

Email mit Type `string`

Passwort mit Type `string`

FK_Ort mit Type `integer`

setmitarbeitermain.md

[Zurück zum Inhaltsverzeichnis](#)

| Name | Type |
|--|--------------|
| PK_Mitarbeiter  | int(11) |
| Vorname | varchar(255) |
| Nachname | varchar(255) |
| Position | varchar(255) |
| Festnetznummer | varchar(20) |
| Mobilnummer | varchar(20) |
| Email | varchar(255) |
| Passwort | varchar(255) |
| FK_Ort  | int(11) |
| FK_Rechtegruppe  | int(11) |

Figure 43: Database Image of Table mitarbeiter

Method: `setmitarbeitermain`

Der Endpoint `setmitarbeitermain` kann in `mitarbeiter` erstellen.

Dieser Endpunkt muss mit folgenden Parametern aufgerufen werden:

`vorname` mit Type `string`

`nachname` mit Type `string`

`position` mit Type `string`

`festnetznummer` mit Type `string`

`mobilnummer` mit Type `string`

`email` mit Type `string`

passwort mit Type string
fk_ort mit Type integer
fk_rechtegruppe mit Type integer
fk_kunde mit Type integer

setortmain.md

[Zurück zum Inhaltsverzeichnis](#)

| Name | Type |
|--|--------------|
| PK_Ort  | int(11) |
| Stadt | varchar(255) |
| PLZ | varchar(10) |
| Straße | varchar(255) |
| Hausnummer | varchar(10) |

Figure 44: Database Image of Table ort

Method: `setortmain`

Der Endpoint `setortmain` kann in `ort` erstellen.

Dieser Endpunkt muss mit folgenden Parametern aufgerufen werden:

Stadt mit Type `string`

PLZ mit Type `string`

Strasse mit Type `string`

Hausnummer mit Type `string`

setprioritaetmain.md

[Zurück zum Inhaltsverzeichnis](#)


| Name | Type |
|--|--------------|
| PK_Priorität  | int(11) |
| Prioritätsname | varchar(255) |
| Beschreibung | text |

Figure 45: Database Image of Table prioritaet

Method: `setprioritaetmain`

Der Endpoint `setprioritaetmain` kann in `prioritaet` erstellen.

Dieser Endpoint muss mit folgenden Parametern aufgerufen werden:

`Prioritaetsname` mit Type `string`

`Beschreibung` mit Type `string`

setrechnungmain.md

[Zurück zum Inhaltsverzeichnis](#)




| Name | Type |
|--|--------------|
| Rechnungsnr  | int(11) |
| Ticketbeschreibung | text |
| Abrechnung | varchar(255) |
| Ratenzahlung | tinyint(1) |
| FK_Bezahlungsart  | int(11) |
| FK_Kunde  | int(11) |
| FK_Ticket  | int(11) |

Figure 46: Database Image of Table rechnung

Method: `setrechnungmain`

Der Endpoint `setrechnungmain` kann in `rechnung` erstellen.

Dieser Endpoint muss mit folgenden Parametern aufgerufen werden:

`Ticketbeschreibung` mit Type `string`

`Abrechnung` mit Type `string`

`Ratenzahlung` mit Type `integer`

`FK_Bezahlungsart` mit Type `integer`

`FK_Kunde` mit Type `integer`

`FK_Ticket` mit Type `integer`

setrechtegruppemain.md

[Zurück zum Inhaltsverzeichnis](#)


| Name | Type |
|--|---------------|
| Administrationsrechte | tinyint(1) |
| Bestelllimit | decimal(10,2) |
| Helpdesk_Fernwartung | tinyint(1) |
| PK_Rechtegruppe  | int(11) |

Figure 47: Database Image of Table rechtegruppe

Method: `setrechtegruppemain`

Der Endpoint `setrechtegruppemain` kann in `rechtegruppe` erstellen.

Dieser Endpunkt muss mit folgenden Parametern aufgerufen werden:

`Administrationsrechte` mit Type `integer`

`Bestelllimit` mit Type `double`

`Helpdesk_Fernwartung` mit Type `integer`

setstatusmain.md

[Zurück zum Inhaltsverzeichnis](#)


| Name | Type |
|---|--------------|
| PK_Status  | int(11) |
| Statusname | varchar(255) |
| Farbe | varchar(20) |
| Beschreibung | text |

Figure 48: Database Image of Table status

Method: `setstatusmain`

Der Endpoint `setstatusmain` kann in `status` erstellen.

Dieser Endpunkt muss mit folgenden Parametern aufgerufen werden:

`Statusname` mit Type `string`

`Farbe` mit Type `string`

`Beschreibung` mit Type `string`

setticketmain.md

[Zurück zum Inhaltsverzeichnis](#)







| Name | Type |
|---|-------------|
| PK_Ticket  | int(11) |
| Ticket_Titel | varchar(75) |
| Beschreibung | text |
| InterneNotiz | text |
| InternerStatus | text |
| Erstelldatum | datetime |
| Bearbeitungsdatum | datetime |
| FK_Status  | int(11) |
| FK_Priorität  | int(11) |
| FK_Art  | int(11) |
| FK_Mitarbeiter  | int(11) |
| FK_Kunde  | int(11) |

Figure 49: Database Image of Table ticket

Method: `setticketmain`

Der Endpoint `setticketmain` kann in `ticket` erstellen.

Dieser Endpoint muss mit folgenden Parametern aufgerufen werden:

`Ticket_Titel` mit Type `string`

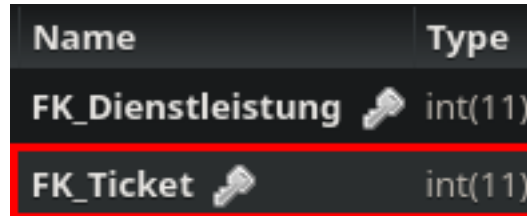
`Beschreibung` mit Type `string`

`InterneNotiz` mit Type `string`

InternerStatus mit Type string
Erstelldatum mit Type string
Bearbeitungsdatum mit Type string
FK_Status mit Type integer
FK_Prioritaet mit Type integer
FK_Art mit Type integer
FK_Mitarbeiter mit Type integer
FK_Kunde mit Type integer
fk_dienstleistung mit Type array

getabgeleitetbytkey.md

[Zurück zum Inhaltsverzeichnis](#)





| Name | Type |
|---|---------|
| FK_Dienstleistung  | int(11) |
| FK_Ticket  | int(11) |

Figure 50: Database Image of Table abgeleitet

Method: `getabgeleitetbytkey`

Der Endpoint `getabgeleitetbytkey` kann in `abgeleitet` holen.

Dieser Endpunkt muss mit folgenden Parametern aufgerufen werden:

`key`(Rot markiert im Bild) mit Type `integer`

getarbeitetbymkey.md

[Zurück zum Inhaltsverzeichnis](#)


| Name | Type |
|--|---------|
| FK_Mitarbeiter  | int(11) |
| FK_Kunde   | int(11) |

Figure 51: Database Image of Table arbeitet

Method: `getarbeitetbymkey`

Der Endpoint `getarbeitetbymkey` kann in `arbeitet` holen.

Dieser Endpunkt muss mit folgenden Parametern aufgerufen werden:

`key`(Rot markiert im Bild) mit Type `integer`

getartall.md

[Zurück zum Inhaltsverzeichnis](#)


| Name | Type |
|--|--------------|
| PK_Art  | int(11) |
| Artname | varchar(255) |
| Beschreibung | text |

Figure 52: Database Image of Table art

Method: `getartall`

Der Endpoint `getartall` kann in `art` holen.

getartbykey.md

[Zurück zum Inhaltsverzeichnis](#)


| Name | Type |
|--|--------------|
| PK_Art  | int(11) |
| Artname | varchar(255) |
| Beschreibung | text |

Figure 53: Database Image of Table art

Method: `getartbykey`

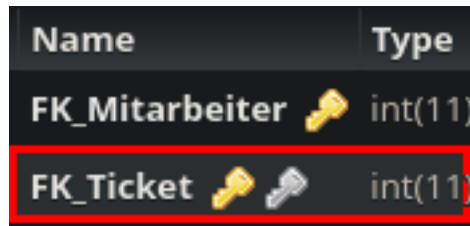
Der Endpoint `getartbykey` kann in `art` holen.

Dieser Endpoint muss mit folgenden Parametern aufgerufen werden:

`key` (Rot markiert im Bild) mit Type `integer`

getbearbeitetbytkey.md

[Zurück zum Inhaltsverzeichnis](#)



| Name | Type |
|----------------|---------|
| FK_Mitarbeiter | int(11) |
| FK_Ticket | int(11) |

Figure 54: Database Image of Table `bearbeitet`

Method: `getbearbeitetbytkey`

Der Endpoint `getbearbeitetbytkey` kann in `bearbeitet` holen.

Dieser Endpoint muss mit folgenden Parametern aufgerufen werden:

`key`(Rot markiert im Bild) mit Type `integer`

getbezahlungsartall.md

[Zurück zum Inhaltsverzeichnis](#)


| Name | Type |
|--|--------------|
| Artname | varchar(255) |
| Beschreibung | text |
| PK_Bezahlungsart  | int(11) |

Figure 55: Database Image of Table bezahlungsart

Method: `getbezahlungsartall`

Der Endpoint `getbezahlungsartall` kann in `bezahlungsart` holen.

getbezahlungsartbykey.md

[Zurück zum Inhaltsverzeichnis](#)


| Name | Type |
|---|--------------|
| Artname | varchar(255) |
| Beschreibung | text |
| PK_Bezahlungsart  | int(11) |

Figure 56: Database Image of Table bezahlungsart

Method: `getbezahlungsartbykey`

Der Endpoint `getbezahlungsartbykey` kann in `bezahlungsart` holen.

Dieser Endpoint muss mit folgenden Parametern aufgerufen werden:

`key` (Rot markiert im Bild) mit Type `integer`

getdienstleistungall.md

[Zurück zum Inhaltsverzeichnis](#)


| Name | Type |
|---|---------------|
| PK_Dienstleistung  | int(11) |
| Beschreibung | text |
| Kosten | decimal(10,2) |

Figure 57: Database Image of Table dienstleistung

Method: `getdienstleistungall`

Der Endpoint `getdienstleistungall` kann in `dienstleistung` holen.

getdienstleistungbykey.md

[Zurück zum Inhaltsverzeichnis](#)


| Name | Type |
|--|---------------|
| PK_Dienstleistung  | int(11) |
| Beschreibung | text |
| Kosten | decimal(10,2) |

Figure 58: Database Image of Table dienstleistung

Method: `getdienstleistungbykey`

Der Endpoint `getdienstleistungbykey` kann in `dienstleistung` holen.

Dieser Endpunkt muss mit folgenden Parametern aufgerufen werden:

`key`(Rot markiert im Bild) mit Type `integer`

getkundeall.md

[Zurück zum Inhaltsverzeichnis](#)


| Name | Type |
|--|--------------|
| PK_Kunde  | int(11) |
| Kundennummer | varchar(255) |
| Firmenname | varchar(255) |
| Email | varchar(255) |
| Passwort | varchar(255) |

Figure 59: Database Image of Table kunde

Method: `getkundeall`

Der Endpoint `getkundeall` kann in `kunde` holen.

getkundebykey.md

[Zurück zum Inhaltsverzeichnis](#)


| Name | Type |
|---|--------------|
| PK_Kunde  | int(11) |
| Kundennummer | varchar(255) |
| Firmenname | varchar(255) |
| Email | varchar(255) |
| Passwort | varchar(255) |

Figure 60: Database Image of Table kunde

Method: `getkundebykey`

Der Endpoint `getkundebykey` kann in `kunde` holen.

Dieser Endpoint muss mit folgenden Parametern aufgerufen werden:

`key`(Rot markiert im Bild) mit Type `integer`

getmitarbeiterall.md

[Zurück zum Inhaltsverzeichnis](#)

| Name | Type |
|--|--------------|
| PK_Mitarbeiter  | int(11) |
| Vorname | varchar(255) |
| Nachname | varchar(255) |
| Position | varchar(255) |
| Festnetznummer | varchar(20) |
| Mobilnummer | varchar(20) |
| Email | varchar(255) |
| Passwort | varchar(255) |
| FK_Ort  | int(11) |
| FK_Rechtegruppe  | int(11) |

Figure 61: Database Image of Table mitarbeiter

Method: `getmitarbeiterall`

Der Endpoint `getmitarbeiterall` kann in `mitarbeiter` holen.

getmitarbeiterbykey.md

[Zurück zum Inhaltsverzeichnis](#)

| Name | Type |
|--|--------------|
| PK_Mitarbeiter  | int(11) |
| Vorname | varchar(255) |
| Nachname | varchar(255) |
| Position | varchar(255) |
| Festnetznummer | varchar(20) |
| Mobilnummer | varchar(20) |
| Email | varchar(255) |
| Passwort | varchar(255) |
| FK_Ort  | int(11) |
| FK_Rechtegruppe  | int(11) |

Figure 62: Database Image of Table mitarbeiter

Method: `getmitarbeiterbykey`

Der Endpoint `getmitarbeiterbykey` kann in `mitarbeiter` holen.

Dieser Endpunkt muss mit folgenden Parametern aufgerufen werden:

`key`(Rot markiert im Bild) mit Type `integer`

getortall.md

[Zurück zum Inhaltsverzeichnis](#)


| Name | Type |
|--|--------------|
| PK_Ort  | int(11) |
| Stadt | varchar(255) |
| PLZ | varchar(10) |
| Straße | varchar(255) |
| Hausnummer | varchar(10) |

Figure 63: Database Image of Table ort

Method: `getortall`

Der Endpoint `getortall` kann in `ort` holen.

getortbykey.md

[Zurück zum Inhaltsverzeichnis](#)

| Name | Type |
|---|--------------|
| PK_Ort  | int(11) |
| Stadt | varchar(255) |
| PLZ | varchar(10) |
| Straße | varchar(255) |
| Hausnummer | varchar(10) |

Figure 64: Database Image of Table ort

Method: `getortbykey`

Der Endpoint `getortbykey` kann in `ort` holen.

Dieser Endpunkt muss mit folgenden Parametern aufgerufen werden:

`key`(Rot markiert im Bild) mit Type `integer`

getprioritaetall.md

[Zurück zum Inhaltsverzeichnis](#)


| Name | Type |
|--|--------------|
| PK_Priorität  | int(11) |
| Prioritätsname | varchar(255) |
| Beschreibung | text |

Figure 65: Database Image of Table prioritaet

Method: `getprioritaetall`

Der Endpoint `getprioritaetall` kann in `prioritaet` holen.

getprioritaetbykey.md

[Zurück zum Inhaltsverzeichnis](#)


| Name | Type |
|--|--------------|
| PK_Priorität  | int(11) |
| Prioritätsname | varchar(255) |
| Beschreibung | text |

Figure 66: Database Image of Table prioritaet

Method: `getprioritaetbykey`

Der Endpoint `getprioritaetbykey` kann in `prioritaet` holen.

Dieser Endpoint muss mit folgenden Parametern aufgerufen werden:

`key` (Rot markiert im Bild) mit Type `integer`

getrechnungall.md

[Zurück zum Inhaltsverzeichnis](#)


| Name | Type |
|--|--------------|
| Rechnungsnr  | int(11) |
| Ticketbeschreibung | text |
| Abrechnung | varchar(255) |
| Ratenzahlung | tinyint(1) |
| FK_Bezahlungsart  | int(11) |
| FK_Kunde  | int(11) |
| FK_Ticket  | int(11) |

Figure 67: Database Image of Table rechnung

Method: `getrechnungall`

Der Endpoint `getrechnungall` kann in `rechnung` holen.

getrechnungbykey.md

[Zurück zum Inhaltsverzeichnis](#)





| Name | Type |
|--|--------------|
| Rechnungsnr  | int(11) |
| Ticketbeschreibung | text |
| Abrechnung | varchar(255) |
| Ratenzahlung | tinyint(1) |
| FK_Bezahlungsart  | int(11) |
| FK_Kunde  | int(11) |
| FK_Ticket  | int(11) |

Figure 68: Database Image of Table rechnung

Method: `getrechnungbykey`

Der Endpoint `getrechnungbykey` kann in `rechnung` holen.

Dieser Endpoint muss mit folgenden Parametern aufgerufen werden:

`key`(Rot markiert im Bild) mit Type `integer`

getrechnungbytkey.md

[Zurück zum Inhaltsverzeichnis](#)





| Name | Type |
|--|--------------|
| Rechnungsnr  | int(11) |
| Ticketbeschreibung | text |
| Abrechnung | varchar(255) |
| Ratenzahlung | tinyint(1) |
| FK_Bezahlungsart  | int(11) |
| FK_Kunde  | int(11) |
| FK_Ticket  | int(11) |

Figure 69: Database Image of Table rechnung

Method: `getrechnungbytkey`

Der Endpoint `getrechnungbytkey` kann in `rechnung` holen.

Dieser Endpoint muss mit folgenden Parametern aufgerufen werden:

`key`(Rot markiert im Bild) mit Type `integer`

getrechtegruppeall.md

[Zurück zum Inhaltsverzeichnis](#)


| Name | Type |
|--|---------------|
| Administrationsrechte | tinyint(1) |
| Bestelllimit | decimal(10,2) |
| Helpdesk_Fernwartung | tinyint(1) |
| PK_Rechtegruppe  | int(11) |

Figure 70: Database Image of Table rechtegruppe

Method: `getrechtegruppeall`

Der Endpoint `getrechtegruppeall` kann in `rechtegruppe` holen.

getrechtegruppebykey.md

[Zurück zum Inhaltsverzeichnis](#)


| Name | Type |
|--|---------------|
| Administrationsrechte | tinyint(1) |
| Bestelllimit | decimal(10,2) |
| Helpdesk_Fernwartung | tinyint(1) |
| PK_Rechtegruppe  | int(11) |

Figure 71: Database Image of Table rechtegruppe

Method: `getrechtegruppebykey`

Der Endpoint `getrechtegruppebykey` kann in `rechtegruppe` holen.

Dieser Endpunkt muss mit folgenden Parametern aufgerufen werden:

`key`(Rot markiert im Bild) mit Type `integer`

getresidiertbykkey.md

[Zurück zum Inhaltsverzeichnis](#)



| Name | Type |
|--|----------|
| FK_Kunde   | int(11) |
| FK_Ort   | int(255) |

Figure 72: Database Image of Table residiert

Method: `getresidiertbykkey`

Der Endpoint `getresidiertbykkey` kann in `residiert` holen.

Dieser Endpunkt muss mit folgenden Parametern aufgerufen werden:

`key`(Rot markiert im Bild) mit Type `integer`

getstatusall.md

[Zurück zum Inhaltsverzeichnis](#)


| Name | Type |
|---|--------------|
| PK_Status  | int(11) |
| Statusname | varchar(255) |
| Farbe | varchar(20) |
| Beschreibung | text |

Figure 73: Database Image of Table status

Method: `getstatusall`

Der Endpoint `getstatusall` kann in `status` holen.

getstatusbykey.md

[Zurück zum Inhaltsverzeichnis](#)


| Name | Type |
|--|--------------|
| PK_Status  | int(11) |
| Statusname | varchar(255) |
| Farbe | varchar(20) |
| Beschreibung | text |

Figure 74: Database Image of Table status

Method: `getstatusbykey`

Der Endpoint `getstatusbykey` kann in `status` holen.

Dieser Endpunkt muss mit folgenden Parametern aufgerufen werden:

`key`(Rot markiert im Bild) mit Type `integer`

getticketall.md

[Zurück zum Inhaltsverzeichnis](#)






| Name | Type |
|---|-------------|
| PK_Ticket  | int(11) |
| Ticket_Titel | varchar(75) |
| Beschreibung | text |
| InterneNotiz | text |
| InternerStatus | text |
| Erstelldatum | datetime |
| Bearbeitungsdatum | datetime |
| FK_Status  | int(11) |
| FK_Priorität  | int(11) |
| FK_Art  | int(11) |
| FK_Mitarbeiter  | int(11) |
| FK_Kunde  | int(11) |

Figure 75: Database Image of Table ticket

Method: `getticketall`

Der Endpoint `getticketall` kann in `ticket` holen.

getticketbykey.md

[Zurück zum Inhaltsverzeichnis](#)





| Name | Type |
|---|-------------|
| PK_Ticket  | int(11) |
| Ticket_Titel | varchar(75) |
| Beschreibung | text |
| InterneNotiz | text |
| InternerStatus | text |
| Erstelldatum | datetime |
| Bearbeitungsdatum | datetime |
| FK_Status  | int(11) |
| FK_Priorität  | int(11) |
| FK_Art  | int(11) |
| FK_Mitarbeiter  | int(11) |
| FK_Kunde  | int(11) |

Figure 76: Database Image of Table ticket

Method: `getticketbykey`

Der Endpoint `getticketbykey` kann in `ticket` holen.

Dieser Endpunkt muss mit folgenden Parametern aufgerufen werden:

`key`(Rot markiert im Bild) mit Type `integer`