

---

# **BuildFlow ERP System**

***Release 1.0.0***

**Enis Seferi**

**Sep 17, 2025**

## API REFERENCE:

<b>1</b>	<b>Features</b>	<b>2</b>
<b>2</b>	<b>Core Application</b>	<b>3</b>
2.1	app.main . . . . .	3
2.2	database.db . . . . .	4
<b>3</b>	<b>Data Models</b>	<b>5</b>
3.1	app.models.kunde . . . . .	5
3.2	app.models.auftrag . . . . .	6
3.3	app.models.rechnung . . . . .	7
3.4	app.models.material_komponente . . . . .	8
3.5	app.models.arbeit_komponente . . . . .	8
3.6	app.models.unternehmensdaten . . . . .	9
3.7	app.models.einnahme_ausgabe . . . . .	10
3.8	app.models.eur_kategorie . . . . .	10
<b>4</b>	<b>API Routes</b>	<b>12</b>
4.1	app.routes.dashboard_route . . . . .	12
4.2	app.routes.kunde_route . . . . .	12
4.3	app.routes.auftrag_route . . . . .	15
4.4	app.routes.rechnung_route . . . . .	16
4.5	app.routes.buchungen . . . . .	19
4.6	app.routes.unternehmensdaten_route . . . . .	21
4.7	app.routes.startseite_route . . . . .	22
4.8	app.routes.auftrag_loeschen . . . . .	23
4.9	app.routes.health . . . . .	23
<b>5</b>	<b>Utilities</b>	<b>24</b>
5.1	app.utils.file_validation . . . . .	24
5.2	app.utils.form_validation . . . . .	24
5.3	app.utils.logging_config . . . . .	26
	<b>Python Module Index</b>	<b>27</b>
	<b>Index</b>	<b>28</b>

BuildFlow is a comprehensive ERP system specifically designed for construction and handcraft businesses. It streamlines business operations from customer management to accounting with an intuitive web interface and guided workflow.

## FEATURES

- **Customer Management** - Private and commercial customers with complete contact information
- **Order Management** - Detailed orders with material and labor components, project tracking
- **Invoice Generation** - Automatic PDF invoice creation with professional formatting
- **Payment Tracking** - Invoice payment status, dates, and receipt uploads
- **Expense Management** - Material cost recording with receipt uploads and manual bookings
- **Accounting (EÜR)** - German tax-compliant income-expense calculation with PDF reports
- **Dashboard & Analytics** - Business overview with revenue tracking and KPIs
- **Health Monitoring** - Production-ready monitoring and logging

## CORE APPLICATION

<code>app.main</code>	FastAPI application for BuildFlow ERP construction management system.
<code>database.db</code>	Database foundation functions and SQLAlchemy setup.

### 2.1 app.main

FastAPI application for BuildFlow ERP construction management system.

#### Functions

<code>from_json(value)</code>	
<code>general_exception_handler(request, exc)</code>	
<code>http_exception_handler(request, exc)</code>	
<code>serve_receipt_file(filename)</code>	Serve receipt files from the receipts directory.
<code>startseite(request[, db])</code>	

```
app.main.from_json(value)
```

```
async app.main.general_exception_handler(request, exc)
```

#### Parameters

- `request` (*Request*)
- `exc` (*Exception*)

```
async app.main.http_exception_handler(request, exc)
```

#### Parameters

- `request` (*Request*)
- `exc` (*HTTPException*)

```
app.main.serve_receipt_file(filename)
```

Serve receipt files from the receipts directory.

**Parameters****filename** (*str*)`app.main.startseite(request, db=Depends(get_db))`**Parameters**

- **request** (*Request*)
- **db** (*Session*)

## 2.2 database.db

Database foundation functions and SQLAlchemy setup.

Provides engine, SessionLocal, Base and FastAPI dependency *get\_db*.

### Functions

`get_db()`

Liefert eine DB-Session für FastAPI-Dependencies.

`database.db.get_db()`

Liefert eine DB-Session für FastAPI-Dependencies.

**Yields**

*Session* – geöffnete SQLAlchemy-Session, wird nach Nutzung geschlossen.

**Return type***Iterator*[*Session*]

## DATA MODELS

<code>app.models.kunde</code>	Customer data models for private and business customers.
<code>app.models.auftrag</code>	Construction order and project management models.
<code>app.models.rechnung</code>	Invoice and billing management models.
<code>app.models.material_komponente</code>	Material cost components for construction orders.
<code>app.models.arbeit_komponente</code>	Work components and labor tracking for construction orders.
<code>app.models.unternehmensdaten</code>	Company data and business information models.
<code>app.models.einnahme_ausgabe</code>	Income and expense tracking for German EÜR tax compliance.
<code>app.models.eur_kategorie</code>	EÜR (German tax) category definitions for income/expense classification.

### 3.1 app.models.kunde

Customer data models for private and business customers.

#### Classes

<code>Kunde(**kwargs)</code>	Customer model for private and business customers.
------------------------------	--

```
class app.models.kunde.Kunde (**kwargs)
```

Bases: Base

Customer model for private and business customers.

Supports both private customers (Privatkunden) and business customers (Gewerbekunden) with conditional fields based on customer type.

**ansprechpartner\_nachname**

**ansprechpartner\_vorname**

**auftraege**

**id**

kunde\_email  
kunde\_firmenname  
kunde\_gesellschaftsform  
kunde\_nachname  
kunde\_rechnung\_ort  
kunde\_rechnung\_plz  
kunde\_rechnungsadresse  
kunde\_seit  
kunde\_telefon  
kunde\_vorname  
kundenart  
notizen  
rechnungen

## 3.2 app.models.auftrag

Construction order and project management models.

### Classes

<i>Auftrag</i> (**kwargs)	Construction order/project model.
---------------------------	-----------------------------------

```
class app.models.auftrag.Auftrag(**kwargs)
```

Bases: Base

Construction order/project model.

Represents a construction job with work and material components, linked to a customer and can have invoices generated.

arbeit\_komponenten

auftrag\_start

beschreibung

id

kunde

kunde\_id

kunde\_leistung\_ort



kunde\_leistung\_plz  
 kunde\_leistungsadresse  
 materialien  
 rechnungen  
 status

### 3.3 app.models.rechnung

Invoice and billing management models.

#### Classes

<i>Rechnung</i> (**kwargs)	Invoice model for billing construction jobs.
----------------------------	--

**class** app.models.rechnung.**Rechnung**(\*\*kwargs)

Bases: Base

Invoice model for billing construction jobs.

Links to customer, order, and company data. Supports payment tracking and EÜR (German tax) entries generation.

auftrag

auftrag\_id

bezahlt

einnahmen\_ausgaben

faelligkeit

id

kunde

kunde\_id

payment\_date

rechnungsdatum

rechnungssumme\_arbeit

rechnungssumme\_gesamt

rechnungssumme\_material

rechtlicher\_hinweis

unternehmensdaten

unternehmensdaten\_id

### 3.4 app.models.material\_komponente

Material cost components for construction orders.

#### Classes

```
MaterialKomponente(**kwargs)
```

```
class app.models.material_komponente.MaterialKomponente (**kwargs)
    Bases: Base

    actual_cost

    anzahl

    auftrag

    auftrag_id

    berechnungseinheit

    bezeichnung

    id

    kategorie

    kategorie_id

    preis_pro_einheit

    receipt_path
```

### 3.5 app.models.arbeit\_komponente

Work components and labor tracking for construction orders.

#### Classes

```
ArbeitKomponente(**kwargs)
```

```
class app.models.arbeit_komponente.ArbeitKomponente (**kwargs)
    Bases: Base

    anzahl_quadrat

    anzahl_stunden

    arbeit
```

`auftrag`  
`auftrag_id`  
`berechnungsbasis`  
`beschreibung`  
`id`  
`kategorie`  
`kategorie_id`  
`komponente_ende`  
`komponente_start`  
`preis_pro_quadrat`  
`stundenlohn`

## 3.6 app.models.unternehmensdaten

Company data and business information models.

### Classes

---

```
Unternehmensdaten(**kwargs)
```

---

```
class app.models.unternehmensdaten.Unternehmensdaten(**kwargs)
```

```
    Bases: Base
```

```
    id
```

```
    rechnungen
```

```
    rechtliche_informationen
```

```
    unternehmen_adresse
```

```
    unternehmen_name
```

```
    unternehmen_ort
```

```
    unternehmen_plz
```

```
    unternehmen_steuernummer
```

```
    unternehmen_telefon
```

```
    zahlungsinfo_bank_name
```

```
    zahlungsinfo_iban
```

`zahlungsinfo_name`  
`zahlungsinfo_paypal`

### 3.7 app.models.einnahme\_ausgabe

Income and expense tracking for German EÜR tax compliance.

#### Classes

```
EinnahmeAusgabe(**kwargs)
```

```
class app.models.einnahme_ausgabe.EinnahmeAusgabe(**kwargs)
    Bases: Base

    beschreibung

    betrag

    datum

    id

    kategorie

    kategorie_id

    receipt_files

    rechnung

    rechnung_id

    typ
```

### 3.8 app.models.eur\_kategorie

EÜR (German tax) category definitions for income/expense classification.

#### Classes

```
EurKategorie(**kwargs)
```

```
class app.models.eur_kategorie.EurKategorie(**kwargs)
    Bases: Base

    arbeit_komponenten
```

einnahmen\_ausgaben

id

material\_komponenten

name

typ

## API ROUTES

<code>app.routes.dashboard_route</code>	
<code>app.routes.kunde_route</code>	Customer management API routes and endpoints.
<code>app.routes.auftrag_route</code>	Order management routes for creating and managing construction projects.
<code>app.routes.rechnung_route</code>	Invoice generation, PDF creation and billing management routes.
<code>app.routes.buchungen</code>	EÜR bookings, income/expense tracking and German tax compliance routes.
<code>app.routes.unternehmensdaten_route</code>	Company data management routes for business information and settings.
<code>app.routes.startseite_route</code>	Home page and main dashboard route.
<code>app.routes.auftrag_loeschen</code>	Order deletion routes with cascading cleanup of related data.
<code>app.routes.health</code>	Health check and system monitoring endpoints.

### 4.1 `app.routes.dashboard_route`

#### Functions

<code>dashboard(request[, year, db])</code>	Business intelligence dashboard with key metrics and charts.
---	--

`app.routes.dashboard_route.dashboard(request, year=None, db=Depends(get_db))`

Business intelligence dashboard with key metrics and charts.

#### Parameters

- **request** (*Request*)
- **year** (*int*)
- **db** (*Session*)

### 4.2 `app.routes.kunde_route`

Customer management API routes and endpoints.

## Functions

<code>formular_anzeigen(request[, db])</code>	Display customer form with list of existing customers.
<code>get_auftrag_details(auftrag_id[, db])</code>	Get detailed auftrag information including components and invoice status.
<code>get_kunde_data(kunde_id[, db])</code>	
<code>kunde_loeschen(kunde_id[, db])</code>	
<code>kunde_speichern(request[, bestehend, ...])</code>	Save new customer or update existing customer data.
<code>kunden_liste(request[, db])</code>	Display customer list with orders and invoice status.
<code>update_auftrag(auftrag_id[, beschreibung, ...])</code>	Update auftrag details (only if no invoice exists).

`app.routes.kunde_route.formular_anzeigen(request, db=Depends(get_db))`

Display customer form with list of existing customers.

### Parameters

- **request** (*Request*)
- **db** (*Session*)

`app.routes.kunde_route.get_auftrag_details(auftrag_id, db=Depends(get_db))`

Get detailed auftrag information including components and invoice status.

### Parameters

- **auftrag\_id** (*int*)
- **db** (*Session*)

`app.routes.kunde_route.get_kunde_data(kunde_id, db=Depends(get_db))`

### Parameters

- **kunde\_id** (*int*)
- **db** (*Session*)

`app.routes.kunde_route.kunde_loeschen(kunde_id, db=Depends(get_db))`

### Parameters

- **kunde\_id** (*int*)
- **db** (*Session*)

```
app.routes.kunde_route.kunde_speichern(request, bestehend=Form(None),
                                       kunde_id=Form(None), kundenart=Form(None),
                                       kunde_firmenname=Form(None),
                                       kunde_gesellschaftsform=Form(None),
                                       ansprechpartner_vorname=Form(None),
                                       ansprechpartner_nachname=Form(None),
                                       kunde_vorname=Form(None),
                                       kunde_nachname=Form(None),
                                       kunde_rechnungsadresse=Form(None),
                                       kunde_rechnung_plz=Form(None),
                                       kunde_rechnung_ort=Form(None),
                                       kunde_email=Form(None),
                                       kunde_telefon=Form(None), notizen=Form(None),
                                       kunde_seit=Form(None), db=Depends(get_db))
```

Save new customer or update existing customer data.

#### Parameters

- **request** (*Request*)
- **bestehend** (*str* | *None*)
- **kunde\_id** (*str* | *None*)
- **kundenart** (*str* | *None*)
- **kunde\_firmenname** (*str* | *None*)
- **kunde\_gesellschaftsform** (*str* | *None*)
- **ansprechpartner\_vorname** (*str* | *None*)
- **ansprechpartner\_nachname** (*str* | *None*)
- **kunde\_vorname** (*str* | *None*)
- **kunde\_nachname** (*str* | *None*)
- **kunde\_rechnungsadresse** (*str* | *None*)
- **kunde\_rechnung\_plz** (*str* | *None*)
- **kunde\_rechnung\_ort** (*str* | *None*)
- **kunde\_email** (*str* | *None*)
- **kunde\_telefon** (*str* | *None*)
- **notizen** (*str* | *None*)
- **kunde\_seit** (*str* | *None*)
- **db** (*Session*)

```
app.routes.kunde_route.kunden_liste(request, db=Depends(get_db))
```

Display customer list with orders and invoice status.

#### Parameters

- **request** (*Request*)
- **db** (*Session*)

```
app.routes.kunde_route.update_auftrag(auftrag_id, beschreibung=Form(PydanticUndefined),
                                       auftrag_start=Form(None),
                                       kunde_leistungsadresse=Form(None),
                                       kunde_leistung_plz=Form(None),
                                       kunde_leistung_ort=Form(None),
                                       arbeitskomponenten=Form(None),
                                       materialkomponenten=Form(None),
                                       db=Depends(get_db))
```

Update auftrag details (only if no invoice exists).



**Parameters**

- `auftrag_id` (*int*)
- `beschreibung` (*str*)
- `auftrag_start` (*str* | *None*)
- `kunde_leistungsadresse` (*str* | *None*)
- `kunde_leistung_plz` (*str* | *None*)
- `kunde_leistung_ort` (*str* | *None*)
- `arbeitskomponenten` (*str* | *None*)
- `materialkomponenten` (*str* | *None*)
- `db` (*Session*)

**4.3 app.routes.auftrag\_route**

Order management routes for creating and managing construction projects.

**Functions**

```
auftrag_anlegen(request[, kunde_id, ...])
```

```
formular_auftrag(request[, db])
```

```
get_kategorie_id(db, name, typ)
```

```
parse_german_decimal(value)
```

Convert German decimal format to float (e.g., '4,5' -> 4.5)

```
async app.routes.auftrag_route.auftrag_anlegen(request,
    kunde_id=Form(PydanticUndefined),
    auftragsbeschrei-
    bung=Form(PydanticUndefined),
    kompo-
    nente_start=Form(PydanticUndefined),
    kompo-
    nente_ende=Form(PydanticUndefined),
    arbeit=Form(PydanticUndefined),
    berechnungsba-
    sis=Form(PydanticUndefined),
    anzahl_stunden=Form(None),
    stundenlohn=Form(None),
    anzahl_quadrat=Form(None),
    preis_pro_quadrat=Form(None),
    beschrei-
    bung_liste=Form(PydanticUndefined),
    material_bezeichnung=Form(None), ma-
    terial_berechnungseinheit=Form(None),
    material_anzahl=Form(None), mate-
    rial_preis_pro_einheit=Form(None),
    kunde_leistungsadresse=Form(PydanticUndefined),
    kunde_leistung_plz=Form(PydanticUndefined),
    kunde_leistung_ort=Form(PydanticUndefined),
    db=Depends(get_db))
```

**Parameters**

- `request` (*Request*)
- `kunde_id` (*int*)
- `auftragsbeschreibung` (*str*)
- `komponente_start` (*List[str]*)
- `komponente_ende` (*List[str]*)
- `arbeit` (*List[str]*)
- `berechnungsbasis` (*List[str]*)
- `anzahl_stunden` (*List[str] | None*)
- `stundenlohn` (*List[str] | None*)
- `anzahl_quadrat` (*List[str] | None*)
- `preis_pro_quadrat` (*List[str] | None*)
- `beschreibung_liste` (*List[str]*)
- `material_bezeichnung` (*List[str] | None*)
- `material_berechnungseinheit` (*List[str] | None*)
- `material_anzahl` (*List[str] | None*)
- `material_preis_pro_einheit` (*List[str] | None*)
- `kunde_leistungsadresse` (*str*)
- `kunde_leistung_plz` (*str*)
- `kunde_leistung_ort` (*str*)
- `db` (*Session*)

`app.routes.auftrag_route.formular_auftrag(request, db=Depends(get_db))`

**Parameters**

- `request` (*Request*)
- `db` (*Session*)

`app.routes.auftrag_route.get_kategorie_id(db, name, typ)`

**Parameters**

- `db` (*Session*)
- `name` (*str*)
- `typ` (*str*)

**Return type**

*int | None*

`app.routes.auftrag_route.parse_german_decimal(value)`

Convert German decimal format to float (e.g., '4,5' -> 4.5)

**Parameters**

`value` (*str | float | int*)

**Return type**

*float*

## 4.4 app.routes.rechnung\_route

Invoice generation, PDF creation and billing management routes.

## Functions

<code>delete_material_receipt(rechnung_id, request)</code>	Delete a specific material receipt file.
<code>delete_rechnung(rechnung_id[, db])</code>	Delete an invoice and all related data.
<code>download_receipt(filename[, db])</code>	Download a receipt file with security validation.
<code>download_rechnung_pdf(rechnung_id[, db])</code>	Download einer spezifischen Rechnungs-PDF.
<code>format_german_decimal(value[, decimals])</code>	Format number with German decimal notation (comma instead of dot)
<code>formular_rechnung(request[, db])</code>	
<code>from_json(value)</code>	
<code>material_kosten_formular(rechnung_id, request)</code>	Form to enter actual material costs for a specific invoice.
<code>material_kosten_speichern(rechnung_id, request)</code>	Save actual material costs and create expense bookings.
<code>rechnung_erstellen(request[, kunde_id, ...])</code>	
<code>rechnungsliste(request[, db, status])</code>	
<code>validate_file_upload(file)</code>	Validate uploaded file for security and constraints.

**async** `app.routes.rechnung_route.delete_material_receipt(rechnung_id, request, db=Depends(get_db))`

Delete a specific material receipt file.

### Parameters

- `rechnung_id` (*int*)
- `request` (*Request*)
- `db` (*Session*)

`app.routes.rechnung_route.delete_rechnung(rechnung_id, db=Depends(get_db))`

Delete an invoice and all related data.

### Parameters

- `rechnung_id` (*int*)
- `db` (*Session*)

`app.routes.rechnung_route.download_receipt(filename, db=Depends(get_db))`

Download a receipt file with security validation.

### Parameters

- `filename` (*str*)
- `db` (*Session*)

`app.routes.rechnung_route.download_rechnung_pdf(rechnung_id, db=Depends(get_db))`

Download einer spezifischen Rechnungs-PDF.

### Parameters

- `rechnung_id` (*int*)
- `db` (*Session*)

```
app.routes.rechnung_route.format_german_decimal(value, decimals=2)
```

Format number with German decimal notation (comma instead of dot)

```
app.routes.rechnung_route.formular_rechnung(request, db=Depends(get_db))
```

#### Parameters

- **request** (*Request*)
- **db** (*Session*)

```
app.routes.rechnung_route.from_json(value)
```

```
app.routes.rechnung_route.material_kosten_formular(rechnung_id, request,  
                                                    db=Depends(get_db))
```

Form to enter actual material costs for a specific invoice.

#### Parameters

- **rechnung\_id** (*int*)
- **request** (*Request*)
- **db** (*Session*)

```
async app.routes.rechnung_route.material_kosten_speichern(rechnung_id, request,  
                                                         db=Depends(get_db))
```

Save actual material costs and create expense bookings.

#### Parameters

- **rechnung\_id** (*int*)
- **request** (*Request*)
- **db** (*Session*)

```
app.routes.rechnung_route.rechnung_erstellen(request,  
                                              kunde_id=Form(PydanticUndefined),  
                                              auftrag_id=Form(PydanticUndefined),  
                                              rechnungsda-  
                                              tum=Form(PydanticUndefined),  
                                              db=Depends(get_db))
```

#### Parameters

- **request** (*Request*)
- **kunde\_id** (*int*)
- **auftrag\_id** (*int*)
- **rechnungsdatum** (*str*)
- **db** (*Session*)

```
app.routes.rechnung_route.rechnungsliste(request, db=Depends(get_db), status=None)
```

#### Parameters

- **request** (*Request*)
- **db** (*Session*)
- **status** (*str*)

```
app.routes.rechnung_route.validate_file_upload(file)
```

Validate uploaded file for security and constraints.

#### Return type

*tuple[bool, str]*

## 4.5 app.routes.buchungen

EÜR bookings, income/expense tracking and German tax compliance routes.

### Functions

<code>delete_manual_booking(booking_id[, db])</code>	
<code>delete_manual_booking_receipt(booking_id[, db])</code>	
<code>edit_manual_booking(booking_id, request[, db])</code>	
<code>euer_uebersicht(request[, view, year, db])</code>	
<code>formular_absenden(request[, betrag, typ, ...])</code>	
<code>formular_anzeigen(request[, db])</code>	
<code>generate_eur_pdf(year[, db])</code>	Generate EÜR PDF report for a specific year
<code>rechnung_status_toggle(rechnung_id[, ...])</code>	
<code>save_manual_booking(booking_id[, datum, ...])</code>	
<code>save_receipt_file(file, booking_id)</code>	Save uploaded file and return the filename.

`app.routes.buchungen.delete_manual_booking(booking_id, db=Depends(get_db))`

#### Parameters

- `booking_id` (*int*)
- `db` (*Session*)

`app.routes.buchungen.delete_manual_booking_receipt(booking_id, filename=Form(PydanticUndefined), db=Depends(get_db))`

#### Parameters

- `booking_id` (*int*)
- `filename` (*str*)
- `db` (*Session*)

`app.routes.buchungen.edit_manual_booking(booking_id, request, db=Depends(get_db))`

#### Parameters

- `booking_id` (*int*)
- `request` (*Request*)
- `db` (*Session*)

`app.routes.buchungen.euer_uebersicht(request, view='list', year=None, db=Depends(get_db))`

#### Parameters

- `request` (*Request*)
- `view` (*str*)
- `year` (*int*)

- `db (Session)`

```
app.routes.buchungen.formular_absenden(request, betrag=Form(PydanticUndefined),
                                         typ=Form(PydanticUndefined),
                                         kategorie_id=Form(PydanticUndefined),
                                         beschreibung=Form(), redirect_to=Form(/euer),
                                         receipt_files=[], datum=Form(None),
                                         bulk_mode=Form(false),
                                         start_datum=Form(None),
                                         end_datum=Form(None), interval=Form(monthly),
                                         db=Depends(get_db))
```

#### Parameters

- `request (Request)`
- `betrag (str)`
- `typ (str)`
- `kategorie_id (str)`
- `beschreibung (str)`
- `redirect_to (str)`
- `receipt_files (List[UploadFile])`
- `datum (str | None)`
- `bulk_mode (str)`
- `start_datum (str | None)`
- `end_datum (str | None)`
- `interval (str)`
- `db (Session)`

```
app.routes.buchungen.formular_anzeigen(request, db=Depends(get_db))
```

#### Parameters

- `request (Request)`
- `db (Session)`

```
app.routes.buchungen.generate_eur_pdf(year, db=Depends(get_db))
```

Generate EÜR PDF report for a specific year

#### Parameters

- `year (int)`
- `db (Session)`

```
app.routes.buchungen.rechnung_status_toggle(rechnung_id, payment_date=Form(None),
                                              status_filter=Form(None),
                                              db=Depends(get_db))
```

#### Parameters

- `rechnung_id (int)`
- `payment_date (str | None)`
- `status_filter (str | None)`
- `db (Session)`

```
app.routes.buchungen.save_manual_booking(booking_id, datum=Form(PydanticUndefined),
                                           betrag=Form(PydanticUndefined),
                                           typ=Form(PydanticUndefined),
                                           kategorie_id=Form(PydanticUndefined),
                                           beschreibung=Form(), receipt_files=[],
                                           db=Depends(get_db))
```

**Parameters**

- `booking_id` (*int*)
- `datum` (*date*)
- `betrag` (*float*)
- `typ` (*str*)
- `kategorie_id` (*int*)
- `beschreibung` (*str*)
- `receipt_files` (*List[UploadFile]*)
- `db` (*Session*)

```
app.routes.buchungen.save_receipt_file(file, booking_id)
```

Save uploaded file and return the filename.

**Parameters**

- `file` (*UploadFile*)
- `booking_id` (*int*)

**Return type**

*str*

## 4.6 app.routes.unternehmensdaten\_route

Company data management routes for business information and settings.

**Functions**

```
unternehmensdaten_formular(request[,  
db])  
unternehmensdaten_speichern(request[,  
...])
```

```
app.routes.unternehmensdaten_route.unternehmensdaten_formular(request,  
db=Depends(get_db))
```

**Parameters**

- `request` (*Request*)
- `db` (*Session*)

```
app.routes.unternehmensdaten_route.unternehmensdaten_speichern(request, un-
    ternehmen_name=Form(PydanticU
un-
    ternehmen_adresse=Form(Pydantic
un-
    ternehmen_plz=Form(PydanticUnd
un-
    ternehmen_ort=Form(PydanticUnd
un-
    ternehmen_steuernummer=Form(Py
un-
    ternehmen_telefon=Form(PydanticU
    zahlungsinfo_name=Form(Pydantic
    zahlungsinfo_bank_name=Form(Py
    zahlungsinfo_iban=Form(PydanticU
    zahlungsinfo_paypal=Form(Pydanti
    rechtliche_informationen=Form(),
    db=Depends(get_db))
```

#### Parameters

- **request** (*Request*)
- **unternehmen\_name** (*str*)
- **unternehmen\_adresse** (*str*)
- **unternehmen\_plz** (*str*)
- **unternehmen\_ort** (*str*)
- **unternehmen\_steuernummer** (*str*)
- **unternehmen\_telefon** (*str*)
- **zahlungsinfo\_name** (*str*)
- **zahlungsinfo\_bank\_name** (*str*)
- **zahlungsinfo\_iban** (*str*)
- **zahlungsinfo\_paypal** (*str*)
- **rechtliche\_informationen** (*str*)
- **db** (*Session*)

## 4.7 app.routes.startseite\_route

Home page and main dashboard route.

### Functions

```
startseite(request)
```

```
app.routes.startseite_route.startseite(request)
```

#### Parameters

- **request** (*Request*)



## 4.8 app.routes.auftrag\_loeschen

Order deletion routes with cascading cleanup of related data.

### Functions

```
auftrag_loeschen(auftrag_id[, db])
```

```
app.routes.auftrag_loeschen.auftrag_loeschen (auftrag_id, db=Depends(get_db))
```

#### Parameters

- **auftrag\_id** (*int*)
- **db** (*Session*)

## 4.9 app.routes.health

Health check and system monitoring endpoints.

### Functions

```
health_check([db])
```

Health check endpoint for monitoring system status.

```
async app.routes.health.health_check (db=Depends(get_db))
```

Health check endpoint for monitoring system status.

#### Parameters

- **db** (*Session*)

<code>app.utils.file_validation</code>	File validation utilities for the ERP system.
<code>app.utils.form_validation</code>	Form validation utilities for the ERP system.
<code>app.utils.logging_config</code>	Production logging configuration with file rotation.

## 5.1 `app.utils.file_validation`

File validation utilities for the ERP system. Consolidates duplicate file validation logic found in `rechnung_route.py` and `buchungen.py`.

### Functions

<code>validate_file_upload(file)</code>	Validate uploaded file for security and constraints.
---	--

`app.utils.file_validation.validate_file_upload(file)`

Validate uploaded file for security and constraints.

#### Parameters

**file** – UploadFile object from FastAPI

#### Returns

(is\_valid, error\_message)

#### Return type

Tuple[bool, str]

## 5.2 `app.utils.form_validation`

Form validation utilities for the ERP system. Complements `file_validation.py` for complete input validation.

### Functions

<code>validate_customer_form(form_data)</code>	Validate customer form and return list of error messages.
--	---

## Classes

<i>FormValidator()</i>	Static methods for validating form input data.
------------------------	--

**class** app.utils.form\_validation.**FormValidator**

Bases: `object`

Static methods for validating form input data.

**static validate\_date** (*date\_str*, *field\_name*, *allow\_future=True*)

Validate date format and logic.

**Parameters**

- **date\_str** (*str*)
- **field\_name** (*str*)
- **allow\_future** (*bool*)

**Return type**

*Tuple*[*bool*, *str*]

**static validate\_email** (*email*)

Validate email format (optional field).

**Parameters**

**email** (*str*)

**Return type**

*Tuple*[*bool*, *str*]

**static validate\_phone** (*phone*)

Validate international phone number format.

**Parameters**

**phone** (*str*)

**Return type**

*Tuple*[*bool*, *str*]

**static validate\_plz** (*plz*)

Validate German postal code.

**Parameters**

**plz** (*str*)

**Return type**

*Tuple*[*bool*, *str*]

**static validate\_positive\_number** (*value*, *field\_name*, *allow\_zero=False*)

Validate positive number.

**Parameters**

- **value** (*str*)
- **field\_name** (*str*)
- **allow\_zero** (*bool*)

**Return type**

*Tuple*[*bool*, *str*]

**static validate\_required\_text** (*text*, *field\_name*, *max\_length=255*)

Validate required text field.

**Parameters**

- **text** (*str*)
- **field\_name** (*str*)
- **max\_length** (*int*)

**Return type***Tuple*[bool, str]

```
app.utils.form_validation.validate_customer_form(form_data)
```

Validate customer form and return list of error messages.

**Parameters****form\_data** (*dict*)**Return type***List*[str]

## 5.3 app.utils.logging\_config

Production logging configuration with file rotation.

### Functions

<code>setup_logging()</code>	Setup basic logging configuration for production
------------------------------	--

```
app.utils.logging_config.setup_logging()
```

Setup basic logging configuration for production

## Python Module Index

### a

`app.main`, 3  
`app.models.arbeit_komponente`, 8  
`app.models.auftrag`, 6  
`app.models.einnahme Ausgabe`, 10  
`app.models.eur_kategorie`, 10  
`app.models.kunde`, 5  
`app.models.material_komponente`, 8  
`app.models.rechnung`, 7  
`app.models.unternehmensdaten`, 9  
`app.routes.auftrag_loeschen`, 23  
`app.routes.auftrag_route`, 15  
`app.routes.buchungen`, 19  
`app.routes.dashboard_route`, 12  
`app.routes.health`, 23  
`app.routes.kunde_route`, 12  
`app.routes.rechnung_route`, 16  
`app.routes.startseite_route`, 22  
`app.routes.unternehmensdaten_route`, 21  
`app.utils.file_validation`, 24  
`app.utils.form_validation`, 24  
`app.utils.logging_config`, 26

### d

`database.db`, 4

# Index

## A

`actual_cost` (*app.models.material\_komponente.MaterialKomponente attribute*), 8  
`ansprechpartner_nachname` (*app.models.kunde.Kunde attribute*), 5  
`ansprechpartner_vorname` (*app.models.kunde.Kunde attribute*), 5  
`anzahl` (*app.models.material\_komponente.MaterialKomponente attribute*), 8  
`anzahl_quadrat` (*app.models.arbeit\_komponente.ArbeitKomponente attribute*), 8  
`anzahl_stunden` (*app.models.arbeit\_komponente.ArbeitKomponente attribute*), 8  
`app.main`  
    module, 3  
`app.models.arbeit_komponente`  
    module, 8  
`app.models.auftrag`  
    module, 6  
`app.models.einnahme_ausgabe`  
    module, 10  
`app.models.eur_kategorie`  
    module, 10  
`app.models.kunde`  
    module, 5  
`app.models.material_komponente`  
    module, 8  
`app.models.rechnung`  
    module, 7  
`app.models.unternehmensdaten`  
    module, 9  
`app.routes.auftrag_loeschen`  
    module, 23  
`app.routes.auftrag_route`  
    module, 15  
`app.routes.buchungen`  
    module, 19  
`app.routes.dashboard_route`  
    module, 12  
`app.routes.health`  
    module, 23  
`app.routes.kunde_route`  
    module, 12  
`app.routes.rechnung_route`  
    module, 16  
`app.routes.startseite_route`  
    module, 22  
`app.routes.unternehmensdaten_route`  
    module, 21  
`app.utils.file_validation`  
    module, 24  
`app.utils.form_validation`  
    module, 24  
`app.utils.logging_config`  
    module, 26  
`arbeit` (*app.models.arbeit\_komponente.ArbeitKomponente attribute*), 8  
`arbeit_komponenten` (*app.models.auftrag.Auftrag attribute*), 6  
`arbeit_komponenten` (*app.models.eur\_kategorie.EurKategorie attribute*), 10  
`ArbeitsKomponente` (class in *app.models.arbeit\_komponente*), 8  
`auftraege` (*app.models.kunde.Kunde attribute*), 5  
`auftrag` (*app.models.arbeit\_komponente.ArbeitKomponente attribute*), 8  
`auftrag` (*app.models.material\_komponente.MaterialKomponente attribute*), 8  
`auftrag` (*app.models.rechnung.Rechnung attribute*), 7  
`Auftrag` (class in *app.models.auftrag*), 6  
`auftrag_anlegen()` (in module *app.routes.auftrag\_route*), 15  
`auftrag_id` (*app.models.arbeit\_komponente.ArbeitKomponente attribute*), 9  
`auftrag_id` (*app.models.material\_komponente.MaterialKomponente attribute*), 8

`auftrag_id` (*app.models.rechnung.Rechnung attribute*), 7  
`auftrag_loeschen()` (in module *app.routes.auftrag\_loeschen*), 23  
`auftrag_start` (*app.models.auftrag.Auftrag attribute*), 6

## B

`berechnungsbasis` (*app.models.arbeit\_komponente.ArbeitKomponente attribute*), 9  
`berechnungseinheit` (*app.models.material\_komponente.MaterialKomponente attribute*), 8  
`beschreibung` (*app.models.arbeit\_komponente.ArbeitKomponente attribute*), 9  
`beschreibung` (*app.models.auftrag.Auftrag attribute*), 6  
`beschreibung` (*app.models.einnahme\_ausgabe.EinnahmeAusgabe attribute*), 10  
`betrag` (*app.models.einnahme\_ausgabe.EinnahmeAusgabe attribute*), 10  
`bezahlt` (*app.models.rechnung.Rechnung attribute*), 7  
`bezeichnung` (*app.models.material\_komponente.MaterialKomponente attribute*), 8

## D

`dashboard()` (in module *app.routes.dashboard\_route*), 12  
`database.db`  
     module, 4  
`datum` (*app.models.einnahme\_ausgabe.EinnahmeAusgabe attribute*), 10  
`delete_manual_booking()` (in module *app.routes.buchungen*), 19  
`delete_manual_booking_receipt()` (in module *app.routes.buchungen*), 19  
`delete_material_receipt()` (in module *app.routes.rechnung\_route*), 17  
`delete_rechnung()` (in module *app.routes.rechnung\_route*), 17  
`download_receipt()` (in module *app.routes.rechnung\_route*), 17  
`download_rechnung_pdf()` (in module *app.routes.rechnung\_route*), 17

## E

`edit_manual_booking()` (in module *app.routes.buchungen*), 19  
`EinnahmeAusgabe` (class in *app.models.einnahme\_ausgabe*), 10  
`einnahmen_ausgaben` (*app.models.eur\_kategorie.EurKategorie attribute*), 10  
`einnahmen_ausgaben` (*app.models.rechnung.Rechnung attribute*), 7  
`euer_uebersicht()` (in module *app.routes.buchungen*), 19  
`EurKategorie` (class in *app.models.eur\_kategorie*), 10

## F

`faelligkeit` (*app.models.rechnung.Rechnung attribute*), 7  
`format_german_decimal()` (in module *app.routes.rechnung\_route*), 17  
`formular_absenden()` (in module *app.routes.buchungen*), 20  
`formular_anzeigen()` (in module *app.routes.buchungen*), 20  
`formular_anzeigen()` (in module *app.routes.kunde\_route*), 13  
`formular_auftrag()` (in module *app.routes.auftrag\_route*), 16  
`formular_rechnung()` (in module *app.routes.rechnung\_route*), 18  
`FormValidator` (class in *app.utils.form\_validation*), 25  
`from_json()` (in module *app.main*), 3  
`from_json()` (in module *app.routes.rechnung\_route*), 18

## G

`general_exception_handler()` (in module *app.main*), 3  
`generate_eur_pdf()` (in module *app.routes.buchungen*), 20  
`get_auftrag_details()` (in module *app.routes.kunde\_route*), 13  
`get_db()` (in module *database.db*), 4  
`get_kategorie_id()` (in module *app.routes.auftrag\_route*), 16  
`get_kunde_data()` (in module *app.routes.kunde\_route*), 13

## H

`health_check()` (in module *app.routes.health*), 23  
`http_exception_handler()` (in module *app.main*), 3

## I

`id` (*app.models.arbeit\_komponente.ArbeitKomponente attribute*), 9  
`id` (*app.models.auftrag.Auftrag attribute*), 6

[id \(app.models.einnahme\\_ausgabe.EinnahmeAusgabe attribute\), 10](#)  
[id \(app.models.eur\\_kategorie.EurKategorie attribute\), 11](#)  
[id \(app.models.kunde.Kunde attribute\), 5](#)  
[id \(app.models.material\\_komponente.MaterialKomponente attribute\), 8](#)  
[id \(app.models.rechnung.Rechnung attribute\), 7](#)  
[id \(app.models.unternehmensdaten.Unternehmensdaten attribute\), 9](#)

## K

[kategorie \(app.models.arbeit\\_komponente.ArbeitKomponente attribute\), 9](#)  
[kategorie \(app.models.einnahme\\_ausgabe.EinnahmeAusgabe attribute\), 10](#)  
[kategorie \(app.models.material\\_komponente.MaterialKomponente attribute\), 8](#)  
[kategorie\\_id \(app.models.arbeit\\_komponente.ArbeitKomponente attribute\), 9](#)  
[kategorie\\_id \(app.models.einnahme\\_ausgabe.EinnahmeAusgabe attribute\), 10](#)  
[kategorie\\_id \(app.models.material\\_komponente.MaterialKomponente attribute\), 8](#)  
[komponente\\_ende \(app.models.arbeit\\_komponente.ArbeitKomponente attribute\), 9](#)  
[komponente\\_start \(app.models.arbeit\\_komponente.ArbeitKomponente attribute\), 9](#)  
[kunde \(app.models.auftrag.Auftrag attribute\), 6](#)  
[kunde \(app.models.rechnung.Rechnung attribute\), 7](#)  
[Kunde \(class in app.models.kunde\), 5](#)  
[kunde\\_email \(app.models.kunde.Kunde attribute\), 5](#)  
[kunde\\_firmenname \(app.models.kunde.Kunde attribute\), 6](#)  
[kunde\\_gesellschaftsform \(app.models.kunde.Kunde attribute\), 6](#)  
[kunde\\_id \(app.models.auftrag.Auftrag attribute\), 6](#)  
[kunde\\_id \(app.models.rechnung.Rechnung attribute\), 7](#)  
[kunde\\_leistung\\_ort \(app.models.auftrag.Auftrag attribute\), 6](#)  
[kunde\\_leistung\\_plz \(app.models.auftrag.Auftrag attribute\), 6](#)  
[kunde\\_leistungsadresse \(app.models.auftrag.Auftrag attribute\), 7](#)  
[kunde\\_loeschen\(\) \(in module app.routes.kunde\\_route\), 13](#)  
[kunde\\_nachname \(app.models.kunde.Kunde attribute\), 6](#)  
[kunde\\_rechnung\\_ort \(app.models.kunde.Kunde attribute\), 6](#)  
[kunde\\_rechnung\\_plz \(app.models.kunde.Kunde attribute\), 6](#)  
[kunde\\_rechnungsadresse \(app.models.kunde.Kunde attribute\), 6](#)  
[kunde\\_seit \(app.models.kunde.Kunde attribute\), 6](#)  
[kunde\\_speichern\(\) \(in module app.routes.kunde\\_route\), 13](#)  
[kunde\\_telefon \(app.models.kunde.Kunde attribute\), 6](#)  
[kunde\\_vorname \(app.models.kunde.Kunde attribute\), 6](#)  
[kunden\\_liste\(\) \(in module app.routes.kunde\\_route\), 14](#)  
[kundenart \(app.models.kunde.Kunde attribute\), 6](#)

## M

[material\\_komponenten \(app.models.eur\\_kategorie.EurKategorie attribute\), 11](#)  
[material\\_kosten\\_formular\(\) \(in module app.routes.rechnung\\_route\), 18](#)  
[material\\_kosten\\_speichern\(\) \(in module app.routes.rechnung\\_route\), 18](#)  
[materialien \(app.models.auftrag.Auftrag attribute\), 7](#)  
[MaterialKomponente \(class in app.models.material\\_komponente\), 8](#)  
[module](#)

- [app.main, 3](#)
- [app.models.arbeit\\_komponente, 8](#)
- [app.models.auftrag, 6](#)
- [app.models.einnahme\\_ausgabe, 10](#)
- [app.models.eur\\_kategorie, 10](#)
- [app.models.kunde, 5](#)
- [app.models.material\\_komponente, 8](#)
- [app.models.rechnung, 7](#)
- [app.models.unternehmensdaten, 9](#)
- [app.routes.auftrag\\_loeschen, 23](#)
- [app.routes.auftrag\\_route, 15](#)
- [app.routes.buchungen, 19](#)
- [app.routes.dashboard\\_route, 12](#)
- [app.routes.health, 23](#)
- [app.routes.kunde\\_route, 12](#)
- [app.routes.rechnung\\_route, 16](#)
- [app.routes.startseite\\_route, 22](#)



[app.routes.unternehmensdaten\\_route](#), 21  
[app.utils.file\\_validation](#), 24  
[app.utils.form\\_validation](#), 24  
[app.utils.logging\\_config](#), 26  
[database.db](#), 4

## N

[name](#) ([app.models.eur\\_kategorie.EurKategorie](#) attribute), 11  
[notizen](#) ([app.models.kunde.Kunde](#) attribute), 6

## P

[parse\\_german\\_decimal\(\)](#) (in module [app.routes.auftrag\\_route](#)), 16  
[payment\\_date](#) ([app.models.rechnung.Rechnung](#) attribute), 7  
[preis\\_pro\\_einheit](#) ([app.models.material\\_komponente.MaterialKomponente](#) attribute), 8  
[preis\\_pro\\_quadrat](#) ([app.models.arbeit\\_komponente.ArbeitKomponente](#) attribute), 9

## R

[receipt\\_files](#) ([app.models.einnahme\\_ausgabe.EinnahmeAusgabe](#) attribute), 10  
[receipt\\_path](#) ([app.models.material\\_komponente.MaterialKomponente](#) attribute), 8  
[rechnung](#) ([app.models.einnahme\\_ausgabe.EinnahmeAusgabe](#) attribute), 10  
[Rechnung](#) (class in [app.models.rechnung](#)), 7  
[rechnung\\_erstellen\(\)](#) (in module [app.routes.rechnung\\_route](#)), 18  
[rechnung\\_id](#) ([app.models.einnahme\\_ausgabe.EinnahmeAusgabe](#) attribute), 10  
[rechnung\\_status\\_toggle\(\)](#) (in module [app.routes.buchungen](#)), 20  
[rechnungen](#) ([app.models.auftrag.Auftrag](#) attribute), 7  
[rechnungen](#) ([app.models.kunde.Kunde](#) attribute), 6  
[rechnungen](#) ([app.models.unternehmensdaten.Unternehmensdaten](#) attribute), 9  
[rechnungsdatum](#) ([app.models.rechnung.Rechnung](#) attribute), 7  
[rechnungsliste\(\)](#) (in module [app.routes.rechnung\\_route](#)), 18  
[rechnungssumme\\_arbeit](#) ([app.models.rechnung.Rechnung](#) attribute), 7  
[rechnungssumme\\_gesamt](#) ([app.models.rechnung.Rechnung](#) attribute), 7  
[rechnungssumme\\_material](#) ([app.models.rechnung.Rechnung](#) attribute), 7  
[rechtliche\\_informationen](#) ([app.models.unternehmensdaten.Unternehmensdaten](#) attribute), 9  
[rechtlicher\\_hinweis](#) ([app.models.rechnung.Rechnung](#) attribute), 7

## S

[save\\_manual\\_booking\(\)](#) (in module [app.routes.buchungen](#)), 20  
[save\\_receipt\\_file\(\)](#) (in module [app.routes.buchungen](#)), 21  
[serve\\_receipt\\_file\(\)](#) (in module [app.main](#)), 3  
[setup\\_logging\(\)](#) (in module [app.utils.logging\\_config](#)), 26  
[startseite\(\)](#) (in module [app.main](#)), 4  
[startseite\(\)](#) (in module [app.routes.startseite\\_route](#)), 22  
[status](#) ([app.models.auftrag.Auftrag](#) attribute), 7  
[stundenlohn](#) ([app.models.arbeit\\_komponente.ArbeitKomponente](#) attribute), 9

## T

[typ](#) ([app.models.einnahme\\_ausgabe.EinnahmeAusgabe](#) attribute), 10  
[typ](#) ([app.models.eur\\_kategorie.EurKategorie](#) attribute), 11

## U

[unternehmen\\_adresse](#) ([app.models.unternehmensdaten.Unternehmensdaten](#) attribute), 9  
[unternehmen\\_name](#) ([app.models.unternehmensdaten.Unternehmensdaten](#) attribute), 9  
[unternehmen\\_ort](#) ([app.models.unternehmensdaten.Unternehmensdaten](#) attribute), 9  
[unternehmen\\_plz](#) ([app.models.unternehmensdaten.Unternehmensdaten](#) attribute), 9  
[unternehmen\\_steuernummer](#) ([app.models.unternehmensdaten.Unternehmensdaten](#) attribute), 9  
[unternehmen\\_telefon](#) ([app.models.unternehmensdaten.Unternehmensdaten](#) attribute), 9  
[unternehmensdaten](#) ([app.models.rechnung.Rechnung](#) attribute), 7  
[Unternehmensdaten](#) (class in [app.models.unternehmensdaten](#)), 9  
[unternehmensdaten\\_formular\(\)](#) (in module [app.routes.unternehmensdaten\\_route](#)), 21  
[unternehmensdaten\\_id](#) ([app.models.rechnung.Rechnung](#) attribute), 7  
[unternehmensdaten\\_speichern\(\)](#) (in module [app.routes.unternehmensdaten\\_route](#)), 21

`update_auftrag()` (in module `app.routes.kunde_route`), 14

## V

`validate_customer_form()` (in module `app.utils.form_validation`), 26  
`validate_date()` (`app.utils.form_validation.FormValidator` static method), 25  
`validate_email()` (`app.utils.form_validation.FormValidator` static method), 25  
`validate_file_upload()` (in module `app.routes.rechnung_route`), 18  
`validate_file_upload()` (in module `app.utils.file_validation`), 24  
`validate_phone()` (`app.utils.form_validation.FormValidator` static method), 25  
`validate_plz()` (`app.utils.form_validation.FormValidator` static method), 25  
`validate_positive_number()` (`app.utils.form_validation.FormValidator` static method), 25  
`validate_required_text()` (`app.utils.form_validation.FormValidator` static method), 25

## Z

`zahlungsinfo_bank_name` (`app.models.unternehmensdaten.Unternehmensdaten` attribute), 9  
`zahlungsinfo_iban` (`app.models.unternehmensdaten.Unternehmensdaten` attribute), 9  
`zahlungsinfo_name` (`app.models.unternehmensdaten.Unternehmensdaten` attribute), 9  
`zahlungsinfo_paypal` (`app.models.unternehmensdaten.Unternehmensdaten` attribute), 10