

A) Interne Reisekostenrichtlinie (BRD)

1. Zweck

Einheitliche, gesetzeskonforme Erstattung dienstlich veranlasster Reisekosten für alle Mitarbeitenden.

2. Rechtsgrundlagen

- **ESTG** — § 3 Nr. 13/16 (steuerfreie Erstattung), § 9 (Werbungskosten)
- **LStR / BMF-Schreiben** — Verpflegungspauschalen, Auslandstabellen
- **UStG** — Vorsteuer nur bei ordnungsgemäßen Rechnungen
- **AO / GoBD** — Belegpflicht, Aufbewahrung
- **DSGVO** — Verarbeitung personenbezogener Reisedaten

3. Erstattungsfähig

Kategorie	Regel
Fahrtkosten	Bahn/ÖPNV/Flug gegen Beleg; PKW nach interner km-Pauschale (z. B. 0,30 €/km)
Übernachtung	Tatsächliche Hotelkosten gegen Rechnung
Verpflegung	Gesetzliche Pauschalen (Inland/Ausland), Kürzung bei gestellten Mahlzeiten
Nebenkosten	Parken, Maut, Gepäck etc. gegen Beleg

4. Nicht erstattungsfähig

Private Ausgaben, Bußgelder, Luxus ohne Freigabe, Kosten ohne dienstlichen Bezug.

5. Prozess

1. Reiseantrag (planbare Reisen) → Freigabe Vorgesetzter
2. Belegsammlung während der Reise
3. Abrechnung im CRM innerhalb von **14 Tagen** nach Reiseende
4. Prüfung & Freigabe durch Vorgesetzten (ROLE_MANAGEMENT)
5. Auszahlung / Export durch Admin (ROLE_ADMIN)
6. Revisionssichere Archivierung im CRM

6. Pflichtangaben

Reisezweck, Zeitraum (Datum/Uhrzeit), Orte, Kostenstelle/Projekt, Einzelpositionen, Mahlzeiten-Kürzungen, Belege.

B) Technische Integration ins CRM

Grundprinzipien

- **Bestehende Rollen** aus RoleHierarchy — keine neuen ROLE_*
- **Twig-MVC** wie VacationController — keine REST-API, keine #[ApiResponse]

- **Integer-Status** wie Vacation (0=offen, 1=genehmigt, 2=abgelehnt, ...)
- **E-Mail** über MsGraphService (Vorlage: VacationEmailService)
- **Navigation** in templates/general/nav.html.twig (neben Urlaub/Zeiterfassung)
- **Übersetzungen** in messages.de.php, messages.en.php, messages.es.php

Rollen-Mapping (bestehend!)

Aktion	Rolle	Analog zu
Eigene Abrechnungen erfassen	ROLE_USER+	vacation_read
Team-Freigaben sehen & genehmigen	ROLE_MANAGEMENT	Urlaub-Freigabe (Nav-Intent)
Export, Stammdaten, Pauschalen pflegen	ROLE_ADMIN	vacation_list / Admin-Funktionen
Vollzugriff	ROLE_SUPER_ADMIN	erbt ROLE_ADMIN

Wichtig: Nav-Sichtbarkeit und #[IsGranted] müssen übereinstimmen (aktuell bei Urlaub leichtes Mismatch Nav=MANAGEMENT vs. Controller=ADMIN).

Dateistruktur (neues Modul)

```
src/Entity/TravelExpense/
    TravelExpenseReport.php      # Kopfdaten, Status, Summen
    TravelExpenseItem.php        # Einzelpositionen
    TravelExpensePerDiem.php     # Pauschalen-Berechnung
src/Controller/TravelExpense/
    TravelExpenseController.php  # Self-Service (CRUD, Einreichen)
src/Form/
    TravelExpenseReportType.php
    TravelExpenseItemType.php
src/Repository/TravelExpense/
    TravelExpenseReportRepository.php
    TravelExpenseItemRepository.php
src/Service/TravelExpense/
    TravelExpenseService.php      # Summen, Validierung
    GermanPerDiemCalculator.php   # Pauschalen nach BMF
    TravelExpenseEmailService.php # Benachrichtigungen
templates/travel_expense/
    read.html.twig               # eigene Abrechnungen (DataTable)
    new.html.twig, edit.html.twig, show.html.twig
    _form.html.twig
    list.html.twig               # Freigabe-Liste (MANAGEMENT)
    _items.html.twig             # Positionen-Partial
```

Datenmodell

TravelExpenseReport (Kopf)

- user → User (Antragsteller)
- project → Project (optional, CRM-Bezug)
- contact → Contact (optional, Kundentermin)
- purpose (Reisezweck)

- start_datetime, end_datetime
- destination (Ort/Land)
- status (integer, wie Vacation)
- total_amount, total_reimbursable
- reg_by, reg_datetime, approved_by, approved_datetime
- rejection_reason (nullable)

TravelExpenseItem (Position)

- report → TravelExpenseReport
- type (transport|lodging|meal_per_diem|other)
- date, description, amount, vat_rate
- kilometers (nullable, für PKW)
- document → Document (Beleg-Upload, bestehendes DMS)
- is_reimbursable (boolean)

TravelExpensePerDiem (Pauschale)

- report, date, hours_absent
- base_amount, meal_breakfast_deduction, meal_lunch_deduction, meal_dinner_deduction
- final_amount, rule_version

Status-Workflow (wie Vacation)

0 = offen (Entwurf / eingereicht)

1 = genehmigt

2 = abgelehnt

3 = von Admin erfasst

4 = storniert

Twig-Filter in AppExtension.php: getTravelExpenseStatus (Kopie von getVacationStatus).

Routes (nur Twig-MVC, keine API)

Route	Name	Methode	Rolle
/travel-expense/read	travel_expense_read	GET	ROLE_USER
/travel-expense/new	travel_expense_new	GET/POST	ROLE_USER
/travel-expense/{id}/edit	travel_expense_edit	GET/POST	ROLE_USER (nur eigene, status=0)
/travel-expense/{id}/show	travel_expense_show	GET	ROLE_USER (eigene) / MANAGEMENT (Team)
/travel-expense/{id}/submit	travel_expense_submit	POST	ROLE_USER
/travel-expense/list	travel_expense_list	GET	ROLE_MANAGEMENT

Route	Name	Methode	Rolle
/travel-expense/{id}/approve	travel_expense_approve	GET	ROLE_MANAGEMENT
/travel-expense/{id}/reject	travel_expense_reject	GET	ROLE_MANAGEMENT
/travel-expense/export	travel_expense_export	GET	ROLE_ADMIN

Freigabe per **GET + Redirect** wie vacation_approve / vacation_reject — kein JSON, kein /api/.

Controller-Muster (aus Vacation übernommen)

```
#[Route(path: '/travel-expense')]
class TravelExpenseController extends AbstractController
{
    #[Route(path: '/read', name: 'travel_expense_read', methods: ['GET'])]
    public function read(...): Response
    {
        $user = $this->getUserBySecurityUser();
        $reports = $reportRepository->findByUser($user);
        return $this->render('travel_expense/read.html.twig', [...]);
    }
    #[IsGranted('ROLE_MANAGEMENT')]
    #[Route(path: '/list', name: 'travel_expense_list', methods: ['GET'])]
    public function list(...): Response { ... }
    #[IsGranted('ROLE_MANAGEMENT')]
    #[Route(path: '/{id}/approve', name: 'travel_expense_approve', methods:
['GET'])]
    public function approve(TravelExpenseReport $report, ...): Response
    {
        $report->setStatus(1);
        $report->setApprovedBy($this->getUserBySecurityUser());
        $report->setApprovedDatetime(new DateTime());
        // E-Mail an Mitarbeiter
        return $this->redirectToRoute('travel_expense_list', [],
Response::HTTP_SEE_OTHER);
    }
}
```

Navigation (nav.html.twig)

Einfügen neben Urlaub (~Zeile 195):

```
{% if is_granted('ROLE_MANAGEMENT') %}
    <a class="dropdown-item" href="{{ path('travel_expense_read') }}">
        <i class="fas fa-receipt"></i> {{ 'lang.travelExpense'|trans }}
    </a>
```

```

    <a class="dropdown-item" href="{ path('travel_expense_list') }">
        <i class="fas fa-check-circle"></i> {{ 'lang.travelExpenseApproval'|trans
    }}
    </a>
{% else %}
    <a class="dropdown-item" href="{ path('travel_expense_read') }">
        <i class="fas fa-receipt"></i> {{ 'lang.travelExpense'|trans }}
    </a>
{% endif %}

```

CRM-Verknüpfungen

Bestehende Entität	Nutzung
User	Antragsteller, reg_by, approved_by
Project	Kostenstelle / Projektbezug
Contact	Kundentermin / Geschäftspartner
Document	Beleg-Upload (bestehendes DMS)
Timerecord	Optional: Reisezeiten aus Zeiterfassung vorbefüllen

Berechnungslogik (GermanPerDiemCalculator)

- Inland: >8h = 14 €, 24h = 28 € (BMF-Stand konfigurierbar)
- Kürzungen Frühstück/Mittag/Abend nach aktueller Vorgabe
- PKW: km × Pauschale (Settings oder Konstante)
- Regelversion pro Abrechnungsdatum speichern (Rechtsänderungen)

E-Mail-Benachrichtigungen (TravelExpenseEmailService)

Ereignis	Empfänger
Abrechnung eingereicht	Vorgesetzter (ROLE_MANAGEMENT)
Genehmigt / Abgelehnt	Antragsteller
Zur Korrektur zurück	Antragsteller

Vorlage: VacationEmailService + templates/email/master.html.twig.

Optional: LexOffice (mod_accounting)

Falls Modul gebucht: genehmigte Abrechnungen als Beleg/Export an LexOffice (LexOfficeController-Muster).

C) Umsetzungsphasen

Phase	Inhalt	Dauer
1 — MVP	Entities, Migration, CRUD, Beleg-Upload, Status-Workflow, Nav, Übersetzungen	2–3 Wochen

Phase	Inhalt	Dauer
2 — Pauschalen	GermanPerDiemCalculator, Mahlzeiten-Kürzungen, E-Mail	1–2 Wochen
3 — Integration	Projekt/Kontakt-Verknüpfung, Export (CSV/PDF), optional LexOffice	1–2 Wochen

D) Was bewusst nicht gebaut wird

- Keine neuen Rollen (ROLE_FINANCE, ROLE_MANAGER, ...)
- Keine API-Routen für Freigaben (POST /api/.../approve)
- Kein Symfony Workflow Component (Integer-Status reicht, wie Vacation)
- Keine Voters (Authorization über #[IsGranted] + Owner-Check im Controller)