

Rapport: Texte français détecté

Total: 1878 occurrences trouvées

Fichiers concernés: 151

LOGGER_FIX_EXPLANATION.md

- **Ligne 1:** # Correction de l'Erreur de Création du Dossier 'logs'
- **Ligne 3:** ## 🔎 Analyse du Problème
- **Ligne 5:** ### Erreur Rencontrée
- **Ligne 12:** ### Localisation du Code Problématique
- **Ligne 14:** - **Problème**: Le logger Winston tentait d'écrire des logs dans des fichiers locaux (logs/error.log , logs/combined.log , logs/exceptions.log , logs/rejections.log) sans vérifier l'existence du
- **Ligne 16:** ### Causes Identifiées
- **Ligne 17:** 1. **Environnement Serverless**: Le chemin /var/task/ indique un environnement serverless (AWS Lambda, Vercel, etc.)
- **Ligne 18:**
 2. **Système de Fichiers en Lecture Seule**: Dans un environnement serverless, le système de fichiers est généralement en lecture seule, sauf pour /tmp``
- **Ligne 19:** 3. **Dossier 'logs' Non Existant**: Aucune vérification n'était faite pour créer le dossier avant d'y écrire
- **Ligne 20:** 4. **File Transports Inappropriés**: L'utilisation de file transports dans un environnement serverless est problématique car :
- **Ligne 21:** - Les fichiers sont éphémères et disparaissent après chaque exécution
- **Ligne 22:** - Le système de fichiers peut être en lecture seule
- **Ligne 23:** - Les logs ne sont pas persistés entre les invocations
- **Ligne 25:** ## ✅ Solution Appliquée
- **Ligne 27:** ### 1. Détection de l'Environnement Serverless
- **Ligne 28:** Ajout de la détection automatique des environnements serverless :
- **Ligne 35:** ### 2. Désactivation Conditionnelle des File Transports
- **Ligne 36:** Les file transports sont maintenant **désactivés automatiquement** en environnement serverless :
- **Ligne 54:** ### 3. Crédit Sécurisé du Dossier Logs
- **Ligne 55:** Pour les environnements locaux/non-serverless, ajout d'une méthode sécurisée de création du dossier :
- **Ligne 70:** **Points clés de cette méthode** :
- **Ligne 71:** - ✅ Utilise recursive: true pour créer les dossiers parents si nécessaire
- **Ligne 72:** - ✅ Vérifie l'existence avec existsSync() avant de créer
- **Ligne 73:** - ✅ Entoure le code d'un try-catch pour gérer les erreurs gracieusement
- **Ligne 74:** - ✅ En cas d'échec, le logger continue de fonctionner avec le console transport

- **Ligne 77:** Les handlers de fichiers pour les exceptions et rejections sont également désactivés en environnement serverless :
- **Ligne 89:** ## 🚧 Comportement Après Correction
- **Ligne 92:** - ✓ **Console Transport uniquement** : Les logs sont envoyés à la console
- **Ligne 93:** - ✓ **Capture par le Service Cloud** : Les logs sont automatiquement capturés par CloudWatch (AWS), Vercel Logs, etc.
- **Ligne 94:** - ✓ **Aucune Erreur** : Plus d'erreur ENOENT lors de la création du dossier
- **Ligne 95:** - ✓ **Performance Optimale** : Pas d'opérations de fichiers inutiles
- **Ligne 97:** ### En Environnement Local/Développement
- **Ligne 98:** - ✓ **Console + File Transports** : Les logs sont à la fois affichés dans la console et sauvegardés dans des fichiers
- **Ligne 99:** - ✓ **Création Automatique du Dossier** : Le dossier logs/ est créé automatiquement s'il n'existe pas
- **Ligne 100:** - ✓ **Rotation des Logs** : Les fichiers de logs sont automatiquement gérés avec rotation (10MB max par fichier)
- **Ligne 102:** ## 📄 Modifications Apportées
- **Ligne 104:** ### Fichier Modifié
- **Ligne 107:** ### Changements Effectués
- **Ligne 108:** 1. Ajout des imports fs et path :
- **Ligne 114:** 2. Ajout de la détection d'environnement serverless (ligne 29)
- **Ligne 116:** 3. Ajout de l'appel à _createLogDirIfNotExist() pour les environnements non-serverless (lignes 31-34)
- **Ligne 118:** 4. Séparation des transports en baseTransports et fileTransports (lignes 36-65)
- **Ligne 120:** 5. Ajout de la méthode privée _createLogDirIfNotExist() (lignes 92-106)
- **Ligne 122:** 6. Désactivation conditionnelle des exception/rejection handlers (lignes 80-88)
- **Ligne 124:** ## 🚀 Déploiement
- **Ligne 126:** Après cette correction, l'application peut être déployée sans erreur dans les environnements suivants :
- **Ligne 132:** - ✓ Environnements locaux (développement)
- **Ligne 134:** ## 📝 Recommandations Supplémentaires
- **Ligne 136:** Pour une solution de logging en production plus robuste, considérez :
- **Ligne 138:**
 2. **Structured Logging** : Le format JSON est déjà activé, facilitant l'analyse des logs
- **Ligne 139:**
 3. **Log Aggregation** : Utiliser un service centralisé pour agréger les logs de toutes les instances
- **Ligne 140:** 4. **Monitoring** : Configurer des alertes sur les erreurs critiques
- **Ligne 142:** ## 💫 Résultat Final
- **Ligne 144:**

L'application est maintenant compatible avec les environnements serverless tout en conservant la fonctionnalité de logging sur fichier en développement local. Le logger s'adapte automatiquement à son

README.md

- **Ligne 194:** 5. **Navigation**: Update lib/menuConfig.ts with new menu items

- **Ligne 245:** - [] Email notifications and templates

app/(dashboard)/(modules)/contracts/simple/page.tsx

- **Ligne 226:** Page {pagination.page} of {pagination.totalPages} • {pagination.total} contract{pagination.total > 1 ? "s" : ""} total

app/(dashboard)/(modules)/expenses/page.tsx

- **Ligne 122:** date: new Date().toISOString().split("T")[0],
- **Ligne 133:** date: new Date().toISOString().split("T")[0],
- **Ligne 143:** description: formData.description,

app/(dashboard)/(modules)/feature-requests/manage/page.tsx

- **Ligne 321:** Date: {format(new Date(request.createdAt), "PPP")}

app/(dashboard)/(modules)/feature-requests/new/page.tsx

- **Ligne 28:** description: z.string().min(10, "Description must be at least 10 characters"),
- **Ligne 321:**
Be as specific as possible. Include current behavior, expected behavior, and use cases.

app/(dashboard)/(modules)/feature-requests/test-tracking/page.tsx

- **Ligne 154:** const page = PLATFORM_PAGES[role as keyof typeof PLATFORM_PAGES]?.find((p) => p.url === pageTitle);
- **Ligne 192:** const filteredPages = (PLATFORM_PAGES[selectedRole as keyof typeof PLATFORM_PAGES] || []).filter((page) =>
- **Ligne 294:** handleCheckboxChange(role, page.url, checked as boolean)

app/(dashboard)/(modules)/invoices/[id]/page.tsx

- **Ligne 136:** marginPaidBy: (data.marginPaidBy || "client") as "client" | "agency" | "contractor",
- **Ligne 383:** description: exp.description,

app/(dashboard)/(modules)/invoices/[id]/page_old.tsx

- **Ligne 147:** marginPaidBy: (data.marginPaidBy || "client") as "client" | "agency" | "contractor",
- **Ligne 437:** Received on \${new Date((data as any).paymentReceivedAt).toLocaleDateString()}``
- **Ligne 460:** Paid on \${new Date((data as any).agencyMarkedPaidAt).toLocaleDateString()}``
- **Ligne 659:** {data.sender?.email && <p className="text-sm">{data.sender.email}</p>}
- **Ligne 686:** {data.receiver?.email && <p className="text-sm">{data.receiver.email}</p>}

- **Ligne 690:** Role: {((data.receiver as any).role.displayName || (data.receiver as any).role.name)}
- **Ligne 695:**
 {(data.receiver as any)?.companies && (data.receiver as any).companyUsers.length > 0 && (
- **Ligne 725:** {(!!(data.receiver as any)?.companies || (data.receiver as any).companyUsers.length === 0) &&
- **Ligne 1097:** <p className="font-medium">{((data as any).sender?.email || "N/A")}</p>
- **Ligne 1122:** <p className="font-medium">{((data as any).receiver?.email || "N/A")}</p>
- **Ligne 1133:** <p className="font-medium">{((data.receiver as any).role.displayName || (data.receiver as any).role.name)}</p>
- **Ligne 1139:** {(data.receiver as any)?.companies && (data.receiver as any).companyUsers.length > 0 && (
- **Ligne 1347:** {(data as any).documents && (data as any).documents.length > 0 ? (

app/(dashboard)/(modules)/invoices/page.tsx

- **Ligne 486:** const isOverdue = new Date(inv.dueDate) < new Date() &&
- **Ligne 726:** const isOverdue = new Date(inv.dueDate) < new Date() &&
- **Ligne 917:** const isOverdue = new Date(inv.dueDate) < new Date() &&

app/(dashboard)/(modules)/leads/page.tsx

- **Ligne 42:** { label: "Total Leads", value: leadStats?.total?.toString() || "0", change: "", icon: TrendingUp },

app/(dashboard)/(modules)/onboarding/page.tsx

- **Ligne 306:** <option value="all">All Status</option>
- **Ligne 307:** <option value="completed">Completed</option>
- **Ligne 308:** <option value="in_progress">In Progress</option>
- **Ligne 309:** <option value="pending_review">Pending Review</option>
- **Ligne 310:** <option value="not_started">Not Started</option>

app/(dashboard)/(modules)/onboarding/templates/page.tsx

- **Ligne 465:** <option value="text">Text Answer</option>
- **Ligne 466:** <option value="file">File Upload</option>

app/(dashboard)/(modules)/payments/payslips/page.tsx

- **Ligne 24:** { value: "2", label: "Février" },
- **Ligne 30:** { value: "8", label: "Août" },
- **Ligne 34:** { value: "12", label: "Décembre" },
- **Ligne 82:** toast.success("Payslip supprimé");
- **Ligne 103:** const email = p.user?.email?.toLowerCase() || "";
- **Ligne 203:** description="Visualisez et gérez les bulletins de paie."
- **Ligne 222:** <p className="text-xs text-gray-500">Ce mois</p>

- **Ligne 228:** <p className="text-xs text-gray-500">Générés</p>
- **Ligne 234:** <p className="text-xs text-gray-500">Envoyés</p>
- **Ligne 265:** description="Créez un bulletin pour commencer."
- **Ligne 266:** actionLabel={canCreate ? "Créer un bulletin" : undefined}
- **Ligne 296:** title="Supprimer le bulletin"
- **Ligne 297:** description="Cette action est définitive."

app/(dashboard)/(modules)/referrals/page.tsx

- **Ligne 123:** description: "Name and email required",
- **Ligne 136:** // === COLUMNS ADAPTÉS À TON NOUVEAU BACKEND ===

app/(dashboard)/(modules)/reports/email-logs/page.tsx

- **Ligne 54:**
<PageHeader title="Email Logs" description="View and monitor all email activity">
- **Ligne 110:**
<EmptyState title="No email logs found" description="No emails match your search criteria" icon={Mail} />
- **Ligne 153:** <p className="text-sm text-gray-600">Page {pagination.page} of {pagination.totalPages}</p>

app/(dashboard)/(modules)/reports/send-email/page.tsx

- **Ligne 94:** <RouteGuard permission="email.access.page">

app/(dashboard)/(modules)/reports/sms-logs/page.tsx

- **Ligne 106:** <EmptyState title="No SMS logs found" description="No SMS messages match your search criteria" icon={MessageSquare} />
- **Ligne 139:** <p className="text-sm text-gray-600">Page {pagination.page} of {pagination.totalPages}</p>

app/(dashboard)/(modules)/reports/user-activity/page.tsx

- **Ligne 29:** const total = activityData?.total || 0

app/(dashboard)/(modules)/settings/branding/login/page.tsx

- **Ligne 41:** description: "Your login page branding has been saved successfully.",
- **Ligne 93:** description="Customize the appearance of your login page for a white-label experience"
- **Ligne 127:** <option value="top">Top Center</option>
- **Ligne 128:** <option value="center">Center</option>
- **Ligne 129:** <option value="left">Top Left</option>

app/(dashboard)/(modules)/settings/companies/page.tsx

- **Ligne 48:** description="Manage your client companies"

app/(dashboard)/(modules)/settings/legal/page.tsx

- **Ligne 87:** description="Manage your organization's terms of service and privacy policy"

app/(dashboard)/(modules)/settings/page.tsx

- **Ligne 41:** description: "Manage payroll service providers",

app/(dashboard)/(modules)/settings/permissions/page.tsx

- **Ligne 171:** {permission.description || 'No description available'}

app/(dashboard)/(modules)/settings/roles/page.tsx

- **Ligne 36:** toast.success("Rôle deleted successfully!")
- **Ligne 42:** toast.error(error?.message || "Failed to delete du rôle")
- **Ligne 61:** return <LoadingState message="Chargement des rôles..." />
- **Ligne 71:** title="Gestion des rôles"
- **Ligne 72:** description="Gérez les rôles et les permissions du système"
- **Ligne 78:** placeholder="Search un rôle..."
- **Ligne 86:** New rôle
- **Ligne 97:** <p className="text-sm text-gray-600">Total des rôles</p>
- **Ligne 219:** title="Delete le rôle"
- **Ligne 220:** description={ Are you sure you want to delete le rôle "\${roleToDelete?.name}" ?
Cette action est irréversible. }

app/(dashboard)/(modules)/settings/tenant/page.tsx

- **Ligne 183:** <option value="Inter">Inter (Default)</option>
- **Ligne 184:** <option value="Roboto">Roboto</option>
- **Ligne 185:** <option value="Open Sans">Open Sans</option>
- **Ligne 186:** <option value="Lato">Lato</option>
- **Ligne 187:** <option value="Montserrat">Montserrat</option>
- **Ligne 188:** <option value="Poppins">Poppins</option>
- **Ligne 189:** <option value="Raleway">Raleway</option>
- **Ligne 190:** <option value="Ubuntu">Ubuntu</option>
- **Ligne 191:** <option value="Nunito">Nunito</option>
- **Ligne 192:** <option value="Playfair Display">Playfair Display</option>
- **Ligne 193:** <option value="Source Sans Pro">Source Sans Pro</option>
- **Ligne 194:** <option value="Merriweather">Merriweather</option>

app/(dashboard)/(modules)/settings/webhooks/page.tsx

- **Ligne 170:** description="Configure webhook endpoints to receive real-time notifications"
- **Ligne 232:** description="Create your first webhook to receive real-time notifications"

app/(dashboard)/(modules)/superadmin/page.tsx

- **Ligne 82:** description={stat.description}

app/(dashboard)/(modules)/tasks/page.tsx

- **Ligne 78:** toast.success("Statut de la tâche mis à jour!")
- **Ligne 82:** toast.error(error.message || "Failed to update de la tâche")
- **Ligne 89:** toast.success("Tâche deleted successfully!")
- **Ligne 94:** toast.error(error.message || "Failed to delete de la tâche")
- **Ligne 124:** return <LoadingState message="Chargement des tâches..." />
- **Ligne 130:** title="Mes Tâches"
- **Ligne 131:** description="Gérez vos tâches assignées et suivez la progression"
- **Ligne 141:** New Tâche
- **Ligne 147:** <StatsCard title="Total des Tâches" value={stats?.total || 0} icon={CheckCircle} />
- **Ligne 175:** title={ Aucune tâche \${activeTab === "pending" ? "en attente" : "terminée"} }
- **Ligne 176:** description={activeTab === "pending" ? "Créez votre première tâche pour commencer" : "Aucune tâche terminée pour le moment"}
- **Ligne 177:** actionLabel={canCreate && activeTab === "pending" ? "New Tâche" : undefined}
- **Ligne 230:** Échéance: {format(new Date(task.dueDate), "dd/MM/yyyy")}
- **Ligne 234:** Assigné par: {task.assignerUser.name || task.assignerUser.email}
- **Ligne 260:** title="Delete la Tâche"
- **Ligne 261:** description="Are you sure you want to delete cette tâche ? Cette action est irréversible."

app/(dashboard)/(modules)/timesheets/[id]/page.tsx

- **Ligne 229:** {new Date(data.startDate).toLocaleDateString()} → {new Date(data.endDate).toLocaleDateString()}
- **Ligne 257:** {new Date(data.startDate).toLocaleDateString()} → {new Date(data.endDate).toLocaleDateString()}
- **Ligne 319:** {new Date(entry.date).toLocaleDateString("en-US", {
- **Ligne 439:** {!canUploadFiles && ((data as any).documents?.length === 0 || !(data as any).documents) && (

app/(dashboard)/(modules)/users/[id]/delegated-access/page.tsx

- **Ligne 125:** description={ Control which users \${user?.name || user?.email} can view and manage }
- **Ligne 186:** new Date(grant.expiresAt) < new Date()

app/(dashboard)/(modules)/users/[id]/page.tsx

- **Ligne 128:** description={ View and manage user details for \${user.email} }

app/(dashboard)/(modules)/users/page.tsx

- **Ligne 55:** toast.success("Utilisateur supprimé avec succès.")
- **Ligne 258:** description={ Are you sure you want to delete "\${userToDelete?.name || userToDelete?.email}" ? This action cannot be undone. }

app/(dashboard)/home/page.tsx

- **Ligne 104:** {stats.contracts.total} total

app/api/auth/[...nextauth]/route.ts

- **Ligne 7:** export { handler as GET, handler as POST }

app/api/auth/generate-reset-token/route.ts

- **Ligne 26:** expiresAt: new Date(Date.now() + 60 * 60 * 1000),

app/api/trpc/[trpc]/route.ts

- **Ligne 13:** export { handler as GET, handler as POST };

app/api/upload/route.ts

- **Ligne 13:** { error: "Non autorisé" },
- **Ligne 67:** { error: Type de fichier non autorisé. Types acceptés: \${allowedTypes.join(", ")} },
- **Ligne 76:** { error: "Le fichier est trop volumineux (max 10MB)" },
- **Ligne 109:** { error: error.message || "Erreur lors de l'upload" },

app/auth/login/page.tsx

- **Ligne 99:** email: account.email,
- **Ligne 145:** <Label htmlFor="email">Email</Label>

app/auth/signin/page.tsx

- **Ligne 117:** <Label htmlFor="email">Email</Label>

components/contracts/shared/DocumentUploader.tsx

- **Ligne 85:** description: description.trim(),
- **Ligne 146:** <Label htmlFor="description">Description *</Label>

components/contracts/simple/AdminReviewModal.tsx

- **Ligne 122:** const formatDate = (date: Date | string): string => {
- **Ligne 123:** const d = typeof date === "string" ? new Date(date) : date;

components/contracts/simple/ContractDocumentViewer.tsx

- **Ligne 76:** date: Date | string | null | undefined
- **Ligne 79:** const d = typeof date === "string" ? new Date(date) : date;

components/contracts/simple/ContractStatusTimeline.tsx

- **Ligne 102:** return date.toLocaleDateString("en-GB", {

components/contracts/simple/ContractorSignatureSection.tsx

- **Ligne 58:** new Date().toISOString().split("T")[0] // Default to today's date
- **Ligne 93:** date: Date | string | null | undefined
- **Ligne 96:** const d = typeof date === "string" ? new Date(date) : date;

components/contracts/simple/CreateNormContractModal.tsx

- **Ligne 129:** if (new Date(formData.startDate) >= new Date(formData.endDate)) {
- **Ligne 130:** errors.push("Invalid dates: Start Date must be before End Date");
- **Ligne 137:** // GROSS → account optional → donc rien ici
- **Ligne 536:** <SelectItem value="client">Client / Agency</SelectItem>

components/contracts/simple/MinimalContractCard.tsx

- **Ligne 89:** const formatDate = (date: Date | string): string => {
- **Ligne 91:** typeof date === "string" ? new Date(date) : date;

components/contracts/simple/MinimalContractView.tsx

- **Ligne 181:** date: Date | string | null | undefined
- **Ligne 185:** typeof date === "string" ? new Date(date) : date;

components/contracts/simple/ModifyContractModal.tsx

- **Ligne 97:** description: description.trim() || undefined,

components/contracts/simple/NormContractView.tsx

- **Ligne 212:** date: Date | string | null | undefined
- **Ligne 216:** typeof date === "string" ? new Date(date) : date;

components/guards/ForbiddenPageContent.tsx

- **Ligne 17:** <CardTitle className="text-2xl">Accès Interdit</CardTitle>
- **Ligne 19:** Vous n'avez pas les permissions nécessaires pour accéder à cette page.
- **Ligne 28:** <Home className="mr-2 h-4 w-4" /> Tableau de bord

components/guards/PageContainer.tsx

- **Ligne 21:** * PageContainer - Conteneur intelligent pour les pages multi-rôles
- **Ligne 23:** * Ce composant adapte le contenu de la page selon les permissions de l'utilisateur.
- **Ligne 24:** * Il permet d'avoir une seule page qui fonctionne différemment pour différents rôles.
- **Ligne 27:** * // Page Invoices qui fonctionne pour Contractor ET Admin
- **Ligne 57:** // Déterminer le mode
- **Ligne 61:** // Mode = "manage" si l'utilisateur a la permission manage
- **Ligne 62:** // Mode = "own" si l'utilisateur a la permission own
- **Ligne 63:** // Mode = "none" si l'utilisateur n'a aucune permission
- **Ligne 73:** ? "Vous êtes en mode administrateur - Vous pouvez voir et gérer toutes les données."
- **Ligne 74:** : "Vous êtes en mode personnel - Vous ne pouvez voir que vos propres données."
- **Ligne 85:** * Hook pour obtenir le mode actuel de la page

components/guards/PermissionGuard.tsx

- **Ligne 25:** alertMessage = "Vous n'avez pas les permissions nécessaires.",

components/guards/index.ts

- **Ligne 3:** * Guards - Composants de protection basés sur les permissions
- **Ligne 5:** * Exportations centralisées des composants de garde.

components/invoices/InvoiceReviewModal.tsx

- **Ligne 120:** marginPaidBy: (data.marginPaidBy || "client") as "client" | "agency" | "contractor",

components/invoices/PendingActions.tsx

- **Ligne 131:** description="You're all caught up! There are no tasks requiring your attention at the moment."

components/invoices/SelfInvoiceDialog.tsx

- **Ligne 302:** {preview.from.email && <p className="text-sm">{preview.from.email}</p>}
- **Ligne 309:** {preview.to.email && <p className="text-sm">{preview.to.email}</p>}

components/invoices/detail/InvoiceMetadata.tsx

- **Ligne 163:** {sender?.email && <p className="text-sm">{sender.email}</p>}
- **Ligne 190:** {receiver?.email && <p className="text-sm">{receiver.email}</p>}

components/layout/header.tsx

- **Ligne 27:** // Capture current page URL and pass it as a query parameter

components/modals/company-modal.tsx

- **Ligne 33:** // (compatibilité totale avec Zod + Prisma)
- **Ligne 261:** Cette company appartient-elle à la plateforme (tenant) ?

components/modals/invoice-modal.tsx

- **Ligne 101:** description: invoice.description ?? "",
- **Ligne 119:** description: li.description,

components/modals/lead-modal.tsx

- **Ligne 36:** email: lead?.email || "",
- **Ligne 89:** email: formData.email,
- **Ligne 146:** <Label htmlFor="email">Email *</Label>

components/modals/payslip-modal.tsx

- **Ligne 36:** { value: 2, label: "Février" },
- **Ligne 42:** { value: 8, label: "Août" },
- **Ligne 46:** { value: 12, label: "Décembre" },
- **Ligne 67:** //  NOTE : Je t'explique plus bas comment ajouter cette route
- **Ligne 148:** toast.success("Payslip créé");
- **Ligne 159:** toast.success("Payslip mis à jour");
- **Ligne 176:** toast.error("Veuillez sélectionner un utilisateur");
- **Ligne 208:** {payslip ? "Modifier le payslip" : "Créer un nouveau payslip"}
- **Ligne 221:** <SelectValue placeholder="Sélectionner un utilisateur" />
- **Ligne 243:** <SelectValue placeholder="Sélectionner un contrat" />
- **Ligne 280:** <Label>Année</Label>
- **Ligne 327:** <Label>Déductions</Label>
- **Ligne 369:** <SelectItem value="generated">Généré</SelectItem>
- **Ligne 370:** <SelectItem value="sent">Envoyé</SelectItem>
- **Ligne 371:** <SelectItem value="paid">Payé</SelectItem>
- **Ligne 390:** <Label>Date de paiement</Label>
- **Ligne 421:** {payslip ? "Enregistrer" : "Créer"}

components/modals/task-modal.tsx

- **Ligne 34:** description: task?.description || "",
- **Ligne 49:** description: task.description || "",
- **Ligne 59:** toast.success("Tâche created successfully!")
- **Ligne 67:** toast.error(error?.message || "Failed to create de la tâche")
- **Ligne 73:** toast.success("Tâche updated successfully!")
- **Ligne 80:** toast.error(error?.message || "Failed to update de la tâche")
- **Ligne 98:** toast.error("Le titre est requis")
- **Ligne 103:** toast.error("Veuillez assigner la tâche à un utilisateur")
- **Ligne 109:** description: formData.description || undefined,
- **Ligne 133:** <DialogTitle>{task ? "Edit Tâche" : "New Tâche"}</DialogTitle>
- **Ligne 136:** ? "Mettez à jour les informations de la tâche."
- **Ligne 137:** : "Remplissez les détails pour créer une nouvelle tâche."
- **Ligne 148:** placeholder="Titre de la tâche"
- **Ligne 154:** <Label htmlFor="description">Description</Label>
- **Ligne 159:** placeholder="Détails de la tâche..."
- **Ligne 166:** <Label htmlFor="assignedTo">Assigner à *</Label>
- **Ligne 172:** <SelectValue placeholder="Select un utilisateur" />
- **Ligne 185:** <Label htmlFor="priority">Priorité</Label>
- **Ligne 191:** <SelectValue placeholder="Select la priorité" />
- **Ligne 204:** <Label htmlFor="dueDate">Date d'échéance</Label>
- **Ligne 219:** {task ? "Mettre à Jour" : "Create"}

components/modals/tenant-modal.tsx

- **Ligne 33:** description: Tenant `\${data.tenant.name}` created with admin \$ {data.user.email}.,

components/modals/user-modal.tsx

- **Ligne 52:** email: user.email,
- **Ligne 64:** toast.success("Utilisateur créé avec succès.")
- **Ligne 71:** toast.error(error?.message || "Erreur lors de la création.")
- **Ligne 77:** toast.success("Utilisateur mis à jour.")
- **Ligne 83:** toast.error(error?.message || "Erreur lors de la mise à jour.")
- **Ligne 100:** if (!formData.name) return toast.error("Le nom est requis.")
- **Ligne 101:** if (!formData.email) return toast.error("L'email est requis.")
- **Ligne 102:** if (!formData.roleId) return toast.error("Le rôle est requis.")
- **Ligne 108:** email: formData.email,
- **Ligne 115:** email: formData.email,
- **Ligne 128:**
 <DialogTitle>{user ? "Modifier un utilisateur" : "Créer un utilisateur"}</DialogTitle>
- **Ligne 131:** ? "Modifiez les informations de cet utilisateur."
- **Ligne 132:** : "Indiquez les détails du nouvel utilisateur."
- **Ligne 152:** <Label htmlFor="email">Email *</Label>

- **Ligne 165:** <Label htmlFor="password">Mot de passe (optionnel)</Label>
- **Ligne 172:** placeholder="Si vide → mot de passe généré"
- **Ligne 179:** <Label>Rôle *</Label>
- **Ligne 186:** <SelectValue placeholder="Sélectionnez un rôle" />
- **Ligne 202:** <Label>Status du compte</Label>
- **Ligne 203:** <p className="text-sm text-muted-foreground">Permettre la connexion</p>
- **Ligne 229:** {user ? "Mettre à jour" : "Créer"}

components/profile/ProfilePage.tsx

- **Ligne 146:** email={user!.email}

components/profile/banks/BankAccountForm.tsx

- **Ligne 118:** <SelectItem key={option.value} value={option.value}>
- **Ligne 137:** <SelectItem key={option.value} value={option.value}>

components/profile/sections/UserSection.tsx

- **Ligne 76:** <Label htmlFor="email">Email Address</Label>
- **Ligne 79:** <Input id="email" type="email" className="pl-9" value={form.email} disabled />

components/providers/trpc-provider.tsx

- **Ligne 13:** <api.Provider client={trpc} queryClient={client}>
- **Ligne 14:** <QueryClientProvider client={client}>

components/remittance/RemittanceDetailsModal.tsx

- **Ligne 125:** <option value="pending">Pending</option>
- **Ligne 126:** <option value="processing">Processing</option>
- **Ligne 127:** <option value="completed">Completed</option>
- **Ligne 128:** <option value="failed">Failed</option>

components/timesheets/TimesheetDetailedTimeline.tsx

- **Ligne 11:** date: Date | string;
- **Ligne 57:** const formatDate = (date: Date | string): string => {
- **Ligne 58:** const d = typeof date === "string" ? new Date(date) : date;
- **Ligne 131:** date: timesheet.sentAt || sentHistory?.createdAt || new Date(),
- **Ligne 167:** events.sort((a, b) => new Date(b.date).getTime() - new Date(a.date).getTime());

components/timesheets/TimesheetDocumentUploader.tsx

- **Ligne 80:** description: description.trim(),
- **Ligne 124:** <Label htmlFor="description">Description *</Label>

components/timesheets/TimesheetReviewModal.tsx

- **Ligne 189:** marginPaidBy: marginPaidBy as "client" | "agency" | "contractor",
- **Ligne 770:** {new Date(entry.date).toLocaleDateString("en-US", {
- **Ligne 832:** Invoice will be sent to: {marginBreakdown?.marginPaidBy === "client" ? "Client" : "Agency"}

components/timesheets/TimesheetStatusTimeline.tsx

- **Ligne 100:** return date.toLocaleDateString("en-US", {

components/timesheets/TimesheetSubmissionForm.tsx

- **Ligne 173:** marginPaidBy: marginPaidBy as "client" | "agency" | "contractor",
- **Ligne 246:** description: Expense receipt: \${expense.category} - \${expense.description} ,
- **Ligne 290:** description: exp.description,

components/ui/action-button.tsx

- **Ligne 53:** // Si pas de permission requise, afficher le bouton normalement

components/ui/context-menu.tsx

- **Ligne 4:** import * as ContextMenuPrimitive from '@radix-ui/react-context-menu';

components/ui/dropdown-menu.tsx

- **Ligne 4:** import * as DropdownMenuPrimitive from '@radix-ui/react-dropdown-menu';

components/ui/navigation-menu.tsx

- **Ligne 2:** import * as NavigationMenuPrimitive from '@radix-ui/react-navigation-menu';
- **Ligne 89:** 'origin-top-center relative mt-1.5 h-[var(--radix-navigation-menu-viewport-height)] w-full overflow-hidden rounded-md border bg-popover text-popover-foreground shadow-lg data-[state=open]:anim

components/ui/task-card.tsx

- **Ligne 58:** {description && <p className="text-sm text-muted-foreground">{description}</p>}

components/workflow/MarginCalculationDisplay.tsx

- **Ligne 20:** date?: Date | string;
- **Ligne 74:** case "client":

hooks/contracts/useContractDocument.ts

- **Ligne 8:** * Hook pour gérer les documents de contrats
- **Ligne 10:** * Fonctionnalités:

- **Ligne 11:** * - uploadSignedVersion: Upload une version signée du contrat
- **Ligne 12:** * - convertFileToBase64: Convertit un fichier en base64
- **Ligne 13:** * - validatePDF: Valide qu'un fichier est bien un PDF
- **Ligne 22:** toast.success("Version signée uploadée avec succès");
- **Ligne 27:** toast.error(error.message || "Échec de l'upload");
- **Ligne 32:** * Convertit un fichier en base64
- **Ligne 42:** throw new Error("Erreur lors de la conversion du fichier");
- **Ligne 49:** * Valide qu'un fichier est un PDF
- **Ligne 52:** // Vérifier le type MIME
- **Ligne 56:** error: "Le fichier doit être un PDF",
- **Ligne 60:** // Vérifier la taille (max 10MB)
- **Ligne 65:** error: "Le fichier ne doit pas dépasser 10MB",
- **Ligne 69:** // Vérifier l'extension
- **Ligne 73:** error: "Le fichier doit avoir l'extension .pdf",
- **Ligne 81:** * Upload une version signée avec validation
- **Ligne 84:** // Valider le fichier
- **Ligne 105:** // L'erreur est déjà gérée par la mutation

hooks/contracts/useNormContract.ts

- **Ligne 7:** * Hook pour gérer les actions spécifiques aux contrats NORM
- **Ligne 10:** * - createNormContract: Créer un nouveau contrat NORM
- **Ligne 11:** * - updateNormContract: Mettre à jour un contrat NORM (draft uniquement)
- **Ligne 12:** * - contractorSignContract: Permettre au contractor de signer le contrat
- **Ligne 20:** toast.success("Contrat NORM créé avec succès");
- **Ligne 21:** // Invalider les queries pour rafraîchir les données
- **Ligne 25:** toast.error(error.message || "Échec de la création du contrat NORM");
- **Ligne 32:** toast.success("Contrat NORM mis à jour avec succès");
- **Ligne 37:** toast.error(error.message || "Échec de la mise à jour du contrat NORM");
- **Ligne 44:** toast.success("Contrat signé avec succès");
- **Ligne 49:** toast.error(error.message || "Échec de la signature du contrat");

hooks/contracts/useSimpleContractWorkflow.ts

- **Ligne 7:** * Hook pour gérer le workflow des contrats simplifiés
- **Ligne 10:** * - submitForReview: Soumet un contrat draft pour review admin
- **Ligne 11:** * - approveContract: Approuve un contrat en pending_admin_review
- **Ligne 12:** * - rejectContract: Rejette un contrat et le remet en draft
- **Ligne 13:** * - activateContract: Active un contrat completed
- **Ligne 14:** * - deleteDraftContract: Supprime un contrat en draft
- **Ligne 22:** toast.success("Contrat soumis pour validation");
- **Ligne 23:** // Invalider les queries pour rafraîchir les données
- **Ligne 28:** toast.error(error.message || "Échec de la soumission");
- **Ligne 35:** toast.success("Contrat approuvé avec succès");
- **Ligne 40:** toast.error(error.message || "Échec de l'approbation");

- **Ligne 47:** toast.success("Contrat rejeté");
- **Ligne 52:** toast.error(error.message || "Échec du rejet");
- **Ligne 59:** toast.success("Contrat activé avec succès");
- **Ligne 64:** toast.error(error.message || "Échec de l'activation");
- **Ligne 71:** toast.success("Contrat supprimé");
- **Ligne 75:** toast.error(error.message || "Échec de la suppression");

hooks/use-debounce.ts

- **Ligne 4:** * Hook pour debouner une valeur
- **Ligne 6:** * @param value - Valeur à debouner
- **Ligne 7:** * @param delay - Délai en ms (défaut: 500ms)
- **Ligne 8:** * @returns Valeur debounçée

hooks/use-permissions.ts

- **Ligne 34:** * ! CORRECT : on utilise status pour savoir si ça charge

hooks/useProfile.ts

- **Ligne 85:** description: "Company information saved successfully.",
- **Ligne 103:** description: "Bank information saved successfully.",
- **Ligne 126:** email: user.email,
- **Ligne 177:** email: user.email,

lib/auth.ts

- **Ligne 19:** email: { label: "Email", type: "email" },
- **Ligne 27:** where: { email: credentials.email, isActive: true },
- **Ligne 33:** email: superAdmin.email,
- **Ligne 47:** where: { email: credentials.email, isActive: true },
- **Ligne 57:** email: user.email,

lib/cache.ts

- **Ligne 119:** auditLogs: (tenantId: string, page: number) => audit:\${tenantId}:page:\${page} ,

lib/config/serviceConfig.ts

- **Ligne 76:** * Check Email service configuration
- **Ligne 93:** /*logger.info('✓ Email Service: Resend configured');*/
- **Ligne 100:** /*logger.info('✓ Email Service: SendGrid configured');*/
- **Ligne 107:** /*logger.info('✓ Email Service: Mailgun configured');*/
- **Ligne 114:** /*logger.info('✓ Email Service: SMTP configured');*/
- **Ligne 123:** '⚠️ Email Service: MOCK MODE - Emails will be logged but not sent. ' +

lib/db.ts

- **Ligne 3:** const globalForPrisma = globalThis as unknown as {

lib/dynamicMenuConfig.ts

- **Ligne 27:** * MENU V3 ● Compatible avec permissions: "resource.action.scope"
- **Ligne 359:** P(Resource.EMAIL, Action.ACCESS, PermissionScope.PAGE),

lib/email/emailService.ts

- **Ligne 3:** * Email Service with Queue Support
- **Ligne 171:** throw new ExternalServiceError('email', 'Invalid email provider');
- **Ligne 241:** // 🔥 NORMALISATION DES ATTACHMENTS POUR SENDGRID
- **Ligne 446:** return (await queueManager.addBulk(QueueNames.EMAIL, jobs)) as any;

lib/email/index.ts

- **Ligne 3:** * Email service exports

lib/errors/AppError.ts

- **Ligne 140:** message || External service \${service} failed ,

lib/feature-request-notifications.ts

- **Ligne 33:** * - Optionally send email notifications based on user preferences

lib/performance.ts

- **Ligne 178:** getMetricsByTimeRange(startDate: Date, endDate: Date): PerformanceMetric[] {

lib/queue/queue.ts

- **Ligne 305:** this.workers.set(queueName, worker as unknown as Worker);
- **Ligne 466:** EMAIL: 'email',

lib/s3.ts

- **Ligne 10:** * DO NOT import this module in client components or pages marked with "use client".
- **Ligne 40:** // si la key inclut déjà le préfixe → ne pas le rajouter

lib/services/MarginCalculationService.ts

- **Ligne 13:** CLIENT = 'client',
- **Ligne 73:** case MarginPaidBy.CLIENT:
- **Ligne 82:** { description: 'Agency margin (paid by client)', amount: marginAmount },

- **Ligne 83:** { description: 'Total invoice to client', amount: totalAmount }
- **Ligne 95:** { description: 'Total invoice to client', amount: totalAmount },
- **Ligne 109:** { description: 'Total invoice to client', amount: totalAmount },
- **Ligne 163:** const marginPaidBy = (contract.marginPaidBy as MarginPaidBy) || MarginPaidBy.CLIENT

lib/services/PaymentWorkflowService.ts

- **Ligne 389:** description: Split payment \${i + 1}/\${splits.length}: \${split.description} ,
- **Ligne 410:** description: Process payment \${i + 1}: \${splitAmount} \${invoice.currency} - \${split.description} ,

lib/services/RemittanceService.ts

- **Ligne 47:** description: input.description,
- **Ligne 116:** description: description || Payment received for invoice ,
- **Ligne 154:** description: description || Payment sent to contractor ,
- **Ligne 191:** description: description || Payment sent to payroll partner ,

lib/sms/smsService.ts

- **Ligne 189:** const message = await client.messages.create({

lib/types.ts

- **Ligne 6:** date: Date

lib/utils.ts

- **Ligne 33:** export function formatDate(date: Date | string | null | undefined): string {
- **Ligne 35:** const d = typeof date === "string" ? new Date(date) : date

lib/validation/schemas.ts

- **Ligne 10:** export const emailSchema = z.string().email('Invalid email address');

lib/workflows/invoice-state-machine.ts

- **Ligne 105:** description: 'Invoice has been sent to client',
- **Ligne 111:** description: 'Agency has marked invoice as paid',

package-lock.json

- **Ligne 29:** "@radix-ui/react-navigation-menu": "1.2.0",
- **Ligne 333:** "resolved": "https://registry.npmjs.org/@aws-sdk/client-s3/-/client-s3-3.927.0.tgz",
- **Ligne 400:** "resolved": "https://registry.npmjs.org/@aws-sdk/client-sso/-/client-sso-3.927.0.tgz",

- **Ligne 1925:** "resolved": "https://registry.npmjs.org/@humanwhocodes/module-importer/-/module-importer-1.0.1.tgz",
- **Ligne 2529:** "resolved": "https://registry.npmjs.org/@prisma/client/-/client-6.7.0.tgz",
- **Ligne 2846:** "resolved": "https://registry.npmjs.org/@radix-ui/react-context-menu/-/react-context-menu-2.2.1.tgz",
- **Ligne 2948:** "resolved": "https://registry.npmjs.org/@radix-ui/react-dropdown-menu/-/react-dropdown-menu-2.1.1.tgz",
- **Ligne 3083:** "resolved": "https://registry.npmjs.org/@radix-ui/react-menu/-/react-menu-2.1.1.tgz",
- **Ligne 3151:** "node_modules/@radix-ui/react-navigation-menu": {
- **Ligne 3153:** "resolved": "https://registry.npmjs.org/@radix-ui/react-navigation-menu/-/react-navigation-menu-1.2.0.tgz",
- **Ligne 3874:** "resolved": "https://registry.npmjs.org/@sendgrid/client/-/client-8.1.6.tgz",
- **Ligne 4313:**
 - "resolved": "https://registry.npmjs.org/@smithy/service-error-classification/-/service-error-classification-4.2.4.tgz",
- **Ligne 4354:** "resolved": "https://registry.npmjs.org/@smithy/smithy-client/-/smithy-client-4.9.2.tgz",
- **Ligne 4708:** "resolved": "https://registry.npmjs.org/@trpc/client/-/client-11.7.1.tgz",
- **Ligne 4968:**
 - "resolved": "https://registry.npmjs.org/@types/hoist-non-react-statics/-/hoist-non-react-statics-3.3.7.tgz",
- **Ligne 6549:** "resolved": "https://registry.npmjs.org/client-only/-/client-only-0.0.1.tgz",
- **Ligne 7660:** "resolved": "https://registry.npmjs.org/date-fns/-/date-fns-3.6.0.tgz",
- **Ligne 7781:** "resolved": "https://registry.npmjs.org/detect-node-es/-/detect-node-es-1.1.0.tgz",
- **Ligne 8006:**
 - "resolved": "https://registry.npmjs.org/es-abstract/-/es-abstract-1.24.0.tgz",
- **Ligne 8074:** "resolved": "https://registry.npmjs.org/es-define-property/-/es-define-property-1.0.1.tgz",
- **Ligne 8082:** "resolved": "https://registry.npmjs.org/es-errors/-/es-errors-1.3.0.tgz",
- **Ligne 8090:**
 - "resolved": "https://registry.npmjs.org/es-iterator-helpers/-/es-iterator-helpers-1.2.1.tgz",
- **Ligne 8117:** "resolved": "https://registry.npmjs.org/es-module-lexer/-/es-module-lexer-1.7.0.tgz",
- **Ligne 8122:** "resolved": "https://registry.npmjs.org/es-object-atoms/-/es-object-atoms-1.1.1.tgz",
- **Ligne 8133:** "resolved": "https://registry.npmjs.org/es-set-tostringtag/-/es-set-tostringtag-2.1.0.tgz",
- **Ligne 8147:** "resolved": "https://registry.npmjs.org/es-shim-unscopables/-/es-shim-unscopables-1.1.0.tgz",
- **Ligne 8159:** "resolved": "https://registry.npmjs.org/es-to-primitive/-/es-to-primitive-1.3.0.tgz",
- **Ligne 9461:**
 - "resolved": "https://registry.npmjs.org/get-symbol-description/-/get-symbol-description-1.1.0.tgz",

- **Ligne 10033:** "resolved": "https://registry.npmjs.org/hoist-non-react-statics/-/hoist-non-react-statics-3.3.2.tgz",
- **Ligne 10369:** "resolved": "https://registry.npmjs.org/is-date-object/-/is-date-object-1.1.0.tgz",
- **Ligne 11070:** "resolved": "https://registry.npmjs.org/lodash-es/-/lodash-es-4.17.21.tgz",
- **Ligne 11947:** "resolved": "https://registry.npmjs.org/openid-client/-/openid-client-5.7.1.tgz",
- **Ligne 13870:** "resolved": "https://registry.npmjs.org/standard-as-callback/-/standard-as-callback-2.1.0.tgz",
- **Ligne 14639:** "resolved": "https://registry.npmjs.org/tiny-case/-/tiny-case-1.0.3.tgz",
- **Ligne 14739:** "resolved": "https://registry.npmjs.org/topojson-client/-/topojson-client-3.1.0.tgz",

package.json

- **Ligne 54:** "@radix-ui/react-navigation-menu": "1.2.0",

scripts/README.md

- **Ligne 51:** - User information (email, name)
- **Ligne 88:** - Total page count

scripts/export-test-pages.ts

- **Ligne 63:** roleCount[page.pageRole] = { total: 0, validated: 0 };
- **Ligne 65:** roleCount[page.pageRole].total++;

scripts/import-requests.ts

- **Ligne 88:** description: request.description,
- **Ligne 97:** createdAt: request.createdAt ? new Date(request.createdAt) : new Date(),
- **Ligne 98:** updatedAt: request.updatedAt ? new Date(request.updatedAt) : new Date(),
- **Ligne 112:** uploadedAt: attachment.uploadedAt ? new Date(attachment.uploadedAt) : new Date(),

scripts/import-test-pages.ts

- **Ligne 82:** testedAt: page.testedAt ? new Date(page.testedAt) : null,
- **Ligne 93:** testedAt: page.testedAt ? new Date(page.testedAt) : null,
- **Ligne 95:** createdAt: page.createdAt ? new Date(page.createdAt) : new Date(),
- **Ligne 101:** console.log(Imported: \${page.pageName} (\${page.pageRole}));
- **Ligne 105:** page: page.pageName || "Unknown",

scripts/seed.ts

- **Ligne 3:** * SEED RBAC V4 - Compatible avec la nouvelle base User-centric
- **Ligne 11:** // IMPORTANT : importer TON nouveau fichier RBAC v4

- **Ligne 79:** // ROLE → PERMISSIONS (clean pour ta DB v4)
- **Ligne 282:** description: perm.description,
- **Ligne 294:** description: perm.description,
- **Ligne 339:** console.log("👤 Création des utilisateurs...");
- **Ligne 381:** where: { tenantId_email: { tenantId, email: u.email } },
- **Ligne 385:** email: u.email,
- **Ligne 395:** console.log("✨ Comptes créés !");
- **Ligne 399:** // SEED DEFAULT CURRENCY + COUNTRY (CORRIGÉ)
- **Ligne 418:** where: { code: "US" }, // ✓ utilise TON champ "code"
- **Ligne 435:** console.log("🏢 Création de la tenant company, compte bancaire et contrats...");
- **Ligne 570:** endDate: new Date(Date.now() + 365 * 24 * 60 * 60 * 1000), // 1 year from now
- **Ligne 595:** console.log("📦 Aucun tenant → création...");
- **Ligne 616:** console.log("✨ Seed terminé !");

server/api/routers/analytics.ts

- **Ligne 99:** gte: new Date(new Date().getFullYear(), new Date().getMonth(), 1),
- **Ligne 203:** trends[date][log.action] = (trends[date][log.action] || 0) + 1;
- **Ligne 263:** lte: new Date(Date.now() + 30 * 24 * 60 * 60 * 1000),

server/api/routers/apiKey.ts

- **Ligne 108:** description: input.description,

server/api/routers/contract.ts

- **Ligne 52:** // 🔥 VALIDATION : Au moins userId OU companyId doit être présent
- **Ligne 58:** message: "Au moins userId ou companyId doit être fourni pour un participant."
- **Ligne 61:** // 🔥 VALIDATION CRITIQUE : Les approvers ne doivent JAMAIS avoir requiresSignature: true
- **Ligne 67:**
message: "Les approvers ne peuvent pas avoir requiresSignature: true. Utilisez le champ 'approved' pour les approbations."
- **Ligne 86:** "pending_approval", // 🔥 Ajouté pour le workflow d'approbation
- **Ligne 87:** "pending_signature", // 🔥 Ajouté pour le workflow de signature
- **Ligne 117:** feePayer: z.string().optional().nullable(), // "client" | "worker" (libre pour l'instant)
- **Ligne 290:** // règles de parentage
- **Ligne 297:** assert(parent!.type === "msa", "Le parent doit être un MSA", "BAD_REQUEST")
- **Ligne 300:** // sécurité : un MSA n'a pas de parent
- **Ligne 301:** assert(!input.parentId, "Un MSA ne peut pas avoir de parent", "BAD_REQUEST")
- **Ligne 336:** requiresSignature: p.role === "approver" ? false : (p.requiresSignature ?? false), // 🔥 Approvers ne peuvent JAMAIS avoir requiresSignature
- **Ligne 337:** approved: false, // 🔥 Initialisé à false, passera à true quand l'approver approuve

- **Ligne 394:** // sécurité : on évite de lier un parent à un MSA
- **Ligne 396:**

```
throw new TRPCError({ code: "BAD_REQUEST", message: "Un MSA ne peut pas avoir de parent" })
```
- **Ligne 408:** // si on modifie parentId sur un SOW → le parent doit rester un MSA
- **Ligne 415:** assert(parent!.type === "msa", "Le parent doit être un MSA", "BAD_REQUEST")
- **Ligne 495:** // autorisations par type OK (procédure couvre les 2 permissions)
- **Ligne 558:** requiresSignature: participant.role === "approver" ? false : (participant.requiresSignature ?? false), // 🔥 Approvers ne peuvent JAMAIS avoir requiresSignature
- **Ligne 559:** approved: false, // 🔥 Initialisé à false, passera à true quand l'approver approuve
- **Ligne 599:** // ACTIONS MÉTIER
- **Ligne 601:** // 1) SEND (GLOBAL) → passe en pending_* et émet notifs
- **Ligne 735:** // à brancher sur ton générateur CSV/Excel/PDF
- **Ligne 765:** documentId: z.string(), // ID du document uploadé
- **Ligne 887:** documentId: z.string(), // ID du document uploadé

server/api/routers/dashboard.ts

- **Ligne 58:** stats.contracts = { total, active, pending, draft: total - active - pending };
- **Ligne 115:** stats.payslips = { total, processed, pending: total - processed };
- **Ligne 138:** stats.users = { total, active, inactive: total - active };

server/api/routers/document.ts

- **Ligne 98:** // 🔥 ICI : on passe download à la fonction S3

server/api/routers/email.ts

- **Ligne 23:** to: z.string().email().or(z.array(z.string().email())),
- **Ligne 98:** description: Sent email: \${input.subject} ,

server/api/routers/emailLog.ts

- **Ligne 74:** hasNext: page < Math.ceil(total / pageSize),
- **Ligne 149:** successRate: total > 0 ? (sent / total) * 100 : 0,

server/api/routers/emailTemplate.ts

- **Ligne 259:** { key: "date", description: "Current date", example: "2025-12-01" },

server/api/routers/expense.ts

- **Ligne 42:** // OWN → on ne voit que nos dépenses
- **Ligne 87:** expenseDate: z.string().refine((v) => !isNaN(Date.parse(v)), "Invalid date format"),
- **Ligne 100:** description: input.description,

server/api/routers/featureRequest.ts

- **Ligne 210:** description: z.string().min(10, "Description must be at least 10 characters"),

server/api/routers/invoice.ts

- **Ligne 175:** // Pour les agences : voir les invoices des contrats où elles sont participantes
- **Ligne 180:** // Trouver tous les contrats où l'utilisateur est une agence
- **Ligne 192:** // Récupérer les invoices de ces contrats
- **Ligne 400:** // OWN → l'utilisateur doit être participant actif du contrat
- **Ligne 435:** description: input.description,
- **Ligne 446:** description: li.description,
- **Ligne 532:** description: input.description,
- **Ligne 541:** description: li.description,
- **Ligne 602:** // Permet à l'agence de marquer une invoice comme payée
- **Ligne 603:** // Crée automatiquement un Payment avec status "pending"
- **Ligne 617:** // 1. Récupérer l'invoice avec le contrat et les participants
- **Ligne 634:** // 2. Vérifier les permissions (si pas admin, vérifier que l'utilisateur est l'agence du contrat)
- **Ligne 638:** // Vérifier que l'utilisateur est une agence participant au contrat
- **Ligne 651:** // 3. Vérifier que l'invoice n'est pas déjà payée
- **Ligne 659:** // 4. Créer un Payment avec status "pending" (sera confirmé par l'admin)
- **Ligne 666:** status: "pending", // En attente de confirmation par l'admin
- **Ligne 682:** // 5. Mettre à jour l'invoice status à "paid"
- **Ligne 696:** // 6. Créer un audit log
- **Ligne 706:** description: "Invoice marked as paid by agency",
- **Ligne 1203:** description: "Invoice marked as paid by agency",
- **Ligne 1401:** description: Work on \${new Date(entry.date).toISOString().slice(0, 10)} (\${entry.hours}h)\${entry.description ? ':' + entry.description : ''},
- **Ligne 1412:** description: Expense: \${expense.title} - \${expense.description || ''},
- **Ligne 1441:** dueDate: new Date(Date.now() + 30 * 24 * 60 * 60 * 1000),
- **Ligne 1497:** description: doc.description,
- **Ligne 1866:** email: contractor?.user?.email || contractor?.company?.contactEmail || "",
- **Ligne 1871:** email: contractorUser.email,
- **Ligne 1894:** description: item.description,
- **Ligne 1910:** dueDate: new Date(Date.now() + 30 * 24 * 60 * 60 * 1000), // 30 days from now
- **Ligne 2156:** dueDate: new Date(Date.now() + 30 * 24 * 60 * 60 * 1000),
- **Ligne 2166:** description: item.description,
- **Ligne 2424:** dueDate: new Date(Date.now() + 30 * 24 * 60 * 60 * 1000),
- **Ligne 2433:** description: item.description,
- **Ligne 2555:** - Email: \${contractor?.user?.email || contractor?.company?.contactEmail || "N/A"}
- **Ligne 2587:** dueDate: new Date(Date.now() + 7 * 24 * 60 * 60 * 1000), // 7 days from now
- **Ligne 2798:** description: p.description,

- **Ligne 2843:** const client = invoice.contract?.participants?.find((p) => p.role === "client");
- **Ligne 2862:** dueDate: new Date(Date.now() + 30 * 24 * 60 * 60 * 1000),

server/api/routers/lead.ts

- **Ligne 81:** email: z.string().email(),
- **Ligne 123:** email: z.string().email().optional(),

server/api/routers/onboarding-template.ts

- **Ligne 84:** description: input.description,
- **Ligne 115:** // on remplace tout, l'id est facultatif/ignoré
- **Ligne 133:** // ⚠ Bon champ : onboardingTemplateId (pas templateId)
- **Ligne 142:** description: input.description,
- **Ligne 173:** // Optionnel: vérif d'usage (users liés)
- **Ligne 180:**
message: "Ce template est assigné à des utilisateurs. Détache-les avant suppression.",
- **Ligne 184:** // Avec onDelete: Cascade, supprimer le template suffit.

server/api/routers/onboarding.ts

- **Ligne 64:** email: u.email,
- **Ligne 72:** progress: total > 0 ? Math.round((approved / total) * 100) : 0,
- **Ligne 337:** message: "Aucun template d'onboarding configuré."

server/api/routers/pageTestStatus.ts

- **Ligne 156:** const percentage = total > 0 ? Math.round((validated / total) * 100) : 0;
- **Ligne 179:** const percentage = total > 0 ? Math.round((validated / total) * 100) : 0;

server/api/routers/payment.ts

- **Ligne 168:** description: input.description,
- **Ligne 183:** // Quand status passe à "completed" → crée automatiquement une Task pour le payroll provider
- **Ligne 232:** // Mettre à jour le paiement
- **Ligne 260:** // ⚡ TRIGGER AUTOMATIQUE : Si status passe à "completed" → créer Task pour payroll provider
- **Ligne 276:** // Créer une Task pour le payroll provider
- **Ligne 295:** dueDate: new Date(Date.now() + 7 * 24 * 60 * 60 * 1000), // 7 jours

server/api/routers/payslip.ts

- **Ligne 48:** // Si tu as un Action.SEND dans ton enum
- **Ligne 67:** // Si le user n'a PAS la permission globale → on limite à ses propres payslips
- **Ligne 136:** // Si le user n'a que OWN → on vérifie qu'il est propriétaire du payslip

- **Ligne 230:** description: Created payslip for \${payslip.user.name ?? payslip.user.email} ,
- **Ligne 293:** description: Updated payslip for \${payslip.user.name ?? payslip.user.email} ,
- **Ligne 336:** description: Deleted payslip for \${payslip.user.name ?? payslip.user.email} ,

server/api/routers/remittance.ts

- **Ligne 24:** // 🔥 Helper pour convertir Decimal en number
- **Ligne 201:** description: input.description || "",
- **Ligne 230:** description: input.description ?? undefined,

server/api/routers/simpleContract.ts

- **Ligne 109:** * Crée un MSA avec upload PDF en une seule étape
- **Ligne 113:** * - Génération automatique du titre depuis le nom du fichier
- **Ligne 114:** * - Création du contrat avec statut "draft"
- **Ligne 115:** * - Création du document lié
- **Ligne 116:** * - Création optionnelle d'un participant company
- **Ligne 127:** // 1. Générer titre depuis filename
- **Ligne 130:** // 2. Créer le contrat MSA (draft)
- **Ligne 150:** // 4. Créer le document lié
- **Ligne 168:** // 4. Trouver la company du user (si existe)
- **Ligne 179:** // 5. Créer un seul participant pour représenter "la partie créatrice"
- **Ligne 180:** // - userId = le user connecté
- **Ligne 192:** // 5b. Créer les participants supplémentaires (si fournis)
- **Ligne 215:** // 7. Récupérer le contrat avec participants
- **Ligne 228:** // 8. Récupérer les documents liés (relation manuelle)
- **Ligne 238:** // 9. Fusionner et retourner le contrat complet
- **Ligne 250:** message: "Échec de la création du MSA",
- **Ligne 262:** * Crée un SOW lié à un MSA parent avec upload PDF
- **Ligne 267:** * - Génération automatique du titre
- **Ligne 268:** * - Création du contrat SOW avec statut "draft"
- **Ligne 269:** * - Héritage des champs du MSA parent (currency, country, etc.)
- **Ligne 270:** * - Création du document lié
- **Ligne 281:** // 1. Valider le MSA parent
- **Ligne 288:** // 2. Générer titre depuis filename
- **Ligne 291:** // 3. Créer le contrat SOW (hériter du parent)
- **Ligne 303:** // Hériter du parent
- **Ligne 317:** // 5. Créer le document lié
- **Ligne 335:** // 6. Créer participant company (optionnel)
- **Ligne 349:** // 6b. Créer les participants supplémentaires (si fournis)
- **Ligne 374:** // 8. Charger les infos du contrat (sans documents)
- **Ligne 398:** // 10. Retourner le contrat complet fusionné
- **Ligne 411:** message: "Échec de la création du SOW",
- **Ligne 439:** // 1 Charger le contrat
- **Ligne 453:** // 2 Vérification OWN

- **Ligne 478:** // 4 Vérifier statut
- **Ligne 486:** // 5 Vérifier main document
- **Ligne 494:** message: "Un document principal doit être uploadé avant soumission",
- **Ligne 498:** // 6 Update du statut
- **Ligne 524:** // 8 Créer historique
- **Ligne 565:** message: "Échec de la soumission pour review",
- **Ligne 577:** * Approuve un contrat en attente de review
- **Ligne 590:** // 1. Charger le contrat (sans documents)
- **Ligne 613:** // 2. Mettre à jour → completed
- **Ligne 629:** // 3. Charger les documents séparément
- **Ligne 639:** // 4. Notifier le créateur
- **Ligne 660:** reason: notes || "Approuvé par admin",
- **Ligne 694:** message: "Échec de l'approbation",
- **Ligne 706:** * Rejette un contrat en attente de review et le remet en draft
- **Ligne 719:** // 1. Récupérer le contrat
- **Ligne 739:** // 3. Update → retour en draft
- **Ligne 753:** // 4. Notification au créateur
- **Ligne 766:** // 5. Historique statut (notes supprimé car n'existe pas)
- **Ligne 773:** reason: "Rejeté par admin",
- **Ligne 774:** // ✗ notes supprimé (n'existe pas dans ton modèle)
- **Ligne 806:** message: "Échec du rejet",
- **Ligne 822:** // 1. Charger le contrat (sans include.documents, car la relation n'existe pas)
- **Ligne 834:** // 2. Valider le statut
- **Ligne 838:** message: "Seuls les contrats completed/active peuvent recevoir une version signée",
- **Ligne 842:** // 3. Récupérer le document principal via findMany
- **Ligne 860:** // 5. Ancienne version -> non latest
- **Ligne 868:** // 6. Créer la nouvelle version signée
- **Ligne 891:** // 7. Mettre à jour le contrat
- **Ligne 926:** message: "Échec de l'upload de la version signée",
- **Ligne 938:** * Active un contrat completed
- **Ligne 951:** // 1. Charger le contrat (sans include.documents)
- **Ligne 970:** message: "Seuls les contrats completed peuvent être activés",
- **Ligne 974:** // 3. Récupérer le(s) documents via findMany()
- **Ligne 987:** [activateContract] Warning: Activation du contrat \${contractId} sans version signée``
- **Ligne 1019:** title: "Contrat activé",
- **Ligne 1020:** message: Le contrat "\${contract.title}" est maintenant actif ,
- **Ligne 1025:** // 6. Historique (notes supprimé car n'existe pas)
- **Ligne 1032:** reason: "Activé par admin",
- **Ligne 1064:** message: "Échec de l'activation",
- **Ligne 1073:**
 - * Permet de mettre à jour le titre et la description d'un contrat MSA/SOW/NORM
- **Ligne 1083:** // 1. Charger le contrat
- **Ligne 1095:** // 2. Construire les données de mise à jour

- **Ligne 1098:** if (description !== undefined) updateData.description = description;
- **Ligne 1100:** // Si rien à mettre à jour
- **Ligne 1108:** // 3. Mettre à jour le contrat
- **Ligne 1140:** message: "Échec de la mise à jour du contrat",
- **Ligne 1172:** // 🚫 SI PAS LIST_GLOBAL → On limite aux contrats où l'user participe
- **Ligne 1247:** hasMore: page * pageSize < total,
- **Ligne 1258:** * Récupère un contrat par son ID avec toutes ses relations
- **Ligne 1265:** * - Historique des statuts
- **Ligne 1274:** // 1 Charger le contrat SANS documents
- **Ligne 1339:** // 2b. Charger les documents partagés (ContractDocuments)
- **Ligne 1368:** // 3 Enrichir le statusHistory pour matcher le front
- **Ligne 1382:** notes: null, // ← champ requis par le front
- **Ligne 1383:** changedByUser: user ?? null // ← ajout calculé
- **Ligne 1401:** message: "Échec de la récupération du contrat",
- **Ligne 1412:** * Supprime un contrat en draft uniquement
- **Ligne 1414:** * Sécurités:
- **Ligne 1415:** * - Seuls les contrats draft peuvent être supprimés
- **Ligne 1416:** * - Les MSA avec SOWs liés ne peuvent pas être supprimés
- **Ligne 1417:** * - Les documents S3 sont supprimés en cascade
- **Ligne 1426:** // 1 Charger le contrat (sans include.documents)
- **Ligne 1439:** // 2 Vérifier statut
- **Ligne 1443:** message: "Seuls les contrats en draft peuvent être supprimés",
- **Ligne 1447:** // 3 Vérifier enfants SOW
- **Ligne 1451:** message: "Impossible de supprimer un MSA qui a des SOWs liés",
- **Ligne 1455:** // 4 Charger les documents associés
- **Ligne 1464:** // 5 Supprimer les fichiers S3 associés
- **Ligne 1476:** // 6 Supprimer les documents de la DB
- **Ligne 1481:** // 7 Supprimer le contrat
- **Ligne 1504:** message: "Contrat supprimé avec succès",
- **Ligne 1511:** message: "Échec de la suppression",
- **Ligne 1774:** * Met à jour un contrat NORM en draft
- **Ligne 1776:** * Seuls les contrats en draft peuvent être modifiés
- **Ligne 1787:** // 1. Charger le contrat
- **Ligne 1799:** // 2. Vérifier que le contrat est en draft
- **Ligne 1803:** message: "Seuls les contrats en draft peuvent être modifiés",
- **Ligne 1807:** // 3. Vérifier que c'est un contrat NORM
- **Ligne 1811:** message: "Seuls les contrats NORM peuvent être mis à jour via cet endpoint",
- **Ligne 1815:** // 4. Préparer les données de mise à jour
- **Ligne 1822:** // Salary type et paiement
- **Ligne 1846:** // 5. Mettre à jour le contrat
- **Ligne 1860:** // 5b. Gérer la mise à jour du participant payroll si nécessaire
- **Ligne 1862:** // Supprimer l'ancien participant payroll
- **Ligne 1870:** // Créer un nouveau participant payroll si payrollUserId est fourni
- **Ligne 1920:** message: "Échec de la mise à jour du contrat NORM",

- **Ligne 1931:** * Permet au contractor de signer son contrat NORM
- **Ligne 1933:** * Met à jour le champ contractorSignedAt
- **Ligne 1944:** // 1. Charger le contrat
- **Ligne 1961:** // 2. Vérifier que c'est un contrat NORM
- **Ligne 1965:** message: "Seuls les contrats NORM peuvent être signés via cet endpoint",
- **Ligne 1969:** // 3. Vérifier que l'utilisateur est le contractor
- **Ligne 1977:** message: "Vous n'êtes pas autorisé à signer ce contrat",
- **Ligne 1981:** // 4. Vérifier que le contrat n'est pas déjà signé
- **Ligne 1985:** message: "Ce contrat a déjà été signé par le contractor",
- **Ligne 1989:** // 5. Mettre à jour la date de signature
- **Ligne 2044:** message: "Échec de la signature du contrat",
- **Ligne 2057:** * Ajouter un participant supplémentaire à un contrat existant.
- **Ligne 2060:** * - contract.update.global : peut ajouter à n'importe quel contrat
- **Ligne 2061:** * - contract.update.own : peut ajouter à ses propres contrats
- **Ligne 2064:** * - Le contrat doit être en draft ou pending
- **Ligne 2065:** * - Au moins userId ou companyId doit être fourni
- **Ligne 2066:** * - L'utilisateur/company doit exister
- **Ligne 2075:** // 1. Vérifier les permissions
- **Ligne 2086:** message: "Vous n'avez pas la permission de modifier ce contrat",
- **Ligne 2090:** // 2. Valider l'ajout du participant
- **Ligne 2093:** // 3. Vérifier si le participant existe déjà
- **Ligne 2106:** message: "Ce participant existe déjà pour ce contrat",
- **Ligne 2110:** // 4. Créer le participant
- **Ligne 2168:** message: "Échec de l'ajout du participant",
- **Ligne 2177:** * Supprimer un participant d'un contrat.
- **Ligne 2180:** * - contract.update.global : peut supprimer de n'importe quel contrat
- **Ligne 2181:** * - contract.update.own : peut supprimer de ses propres contrats
- **Ligne 2184:**
 - * - Les participants principaux (company_tenant, agency, contractor) ne peuvent pas être supprimés
- **Ligne 2185:** * - Le contrat doit être en draft ou pending
- **Ligne 2194:** // 1. Récupérer le participant
- **Ligne 2214:** // 2. Vérifier les permissions
- **Ligne 2225:** message: "Vous n'avez pas la permission de modifier ce contrat",
- **Ligne 2229:** // 3. Vérifier que le contrat n'est pas completed/active
- **Ligne 2236:**
 - message: "Impossible de supprimer des participants d'un contrat complété ou actif",
- **Ligne 2240:** // 4. Vérifier que ce n'est pas un participant principal
- **Ligne 2244:** message: "Impossible de supprimer un participant principal (company_tenant, agency, contractor)",
- **Ligne 2248:** // 5. Supprimer le participant
- **Ligne 2272:** message: "Participant supprimé avec succès",
- **Ligne 2279:** message: "Échec de la suppression du participant",
- **Ligne 2288:** * Lister tous les participants d'un contrat.
- **Ligne 2291:** * - contract.read.global : peut lister les participants de tous les contrats

- **Ligne 2292:** * - contract.read.own : peut lister les participants de ses contrats
- **Ligne 2301:** // 1. Vérifier que l'utilisateur peut voir ce contrat
- **Ligne 2312:** message: "Vous n'avez pas la permission de voir ce contrat",
- **Ligne 2316:** // 2. Récupérer tous les participants
- **Ligne 2355:** message: "Échec de la récupération des participants",
- **Ligne 2368:** * Uploader un document partagé pour un contrat.
- **Ligne 2369:** * Tous les participants peuvent uploader des documents.
- **Ligne 2372:** * - Être participant du contrat
- **Ligne 2373:** * - Le contrat ne doit pas être "completed" ou "active"
- **Ligne 2374:** * - Exception: contract.update.global peut toujours uploader
- **Ligne 2383:** // 1. Vérifier que l'utilisateur peut uploader
- **Ligne 2394:** message: "Vous n'avez pas la permission d'uploader des documents pour ce contrat",
- **Ligne 2398:** // 2. Vérifier que le contrat existe
- **Ligne 2415:** // 3. Upload du fichier vers S3
- **Ligne 2421:** // 4. Créer l'entrée Document
- **Ligne 2438:** // 5. Créer l'entrée ContractDocument
- **Ligne 2496:** message: "Échec de l'upload du document",
- **Ligne 2505:** * Lister tous les documents partagés d'un contrat.
- **Ligne 2506:** * Tous les participants peuvent voir les documents.
- **Ligne 2509:** * - Être participant du contrat OU avoir contract.read.global
- **Ligne 2518:** // 1. Vérifier que l'utilisateur peut voir ce contrat
- **Ligne 2529:** message: "Vous n'avez pas la permission de voir ce contrat",
- **Ligne 2533:** // 2. Récupérer tous les documents
- **Ligne 2571:** message: "Échec de la récupération des documents",
- **Ligne 2580:** * Supprimer un document partagé.
- **Ligne 2581:** * Seul l'uploader ou un admin (contract.update.global) peut supprimer.
- **Ligne 2584:** * - Être l'uploader du document OU avoir contract.update.global
- **Ligne 2585:** * - Le contrat ne doit pas être "completed" ou "active"
- **Ligne 2594:** // 1. Récupérer le document
- **Ligne 2621:** // 2. Vérifier les permissions
- **Ligne 2632:** message: "Vous n'avez pas la permission de supprimer ce document",
- **Ligne 2636:** // 3. Supprimer le fichier de S3
- **Ligne 2643:** // 4. Supprimer l'entrée Document en premier
- **Ligne 2648:** // 5. Supprimer l'entrée ContractDocument ensuite
- **Ligne 2674:** message: "Document supprimé avec succès",
- **Ligne 2681:** message: "Échec de la suppression du document",
- **Ligne 2690:** * Obtenir l'URL signée pour télécharger un document.
- **Ligne 2691:** * Tous les participants peuvent télécharger les documents.
- **Ligne 2694:** * - Être participant du contrat OU avoir contract.read.global
- **Ligne 2703:** // 1. Récupérer le document
- **Ligne 2725:** // 2. Vérifier que l'utilisateur peut voir ce contrat
- **Ligne 2736:** message: "Vous n'avez pas la permission de télécharger ce document",

- **Ligne**

- **Ligne 2740:** // 3. Générer l'URL signée (utiliser la fonction existante ou générer manuellement)
- **Ligne 2741:** // Pour l'instant, on retourne juste les infos du document
- **Ligne 2742:** // Le frontend utilisera document.getSignedUrl avec l'ID du document
- **Ligne 2758:** message: "Échec de la récupération du document",
- **Ligne 2771:** * Récupère la company associée à un utilisateur.
- **Ligne 2772:** * Utile pour la fonctionnalité "lier la company du user".
- **Ligne 2775:** * - Accessible à tous les utilisateurs authentifiés
- **Ligne 2785:** // Chercher une CompanyUser active pour cet utilisateur
- **Ligne 2802:** createdAt: "desc", // Prendre la plus récente si plusieurs
- **Ligne 2814:** message: "Échec de la récupération de la company",

server/api/routers/smsLog.ts

- **Ligne 82:** hasNext: page < Math.ceil(total / pageSize),
- **Ligne 149:** successRate: total > 0 ? (sent / total) * 100 : 0,

server/api/routers/tag.ts

- **Ligne 25:** isActive: z.boolean().optional(), // ⚠ ton modèle n'a pas isActive, donc je ne filtre pas dessus

server/api/routers/tenant.ts

- **Ligne 251:** description: Tenant créé avec admin \${input.adminEmail} ,
- **Ligne 355:**
subscriptionEndDate: new Date(Date.now() + 30 * 24 * 60 * 60 * 1000), // 30 days from now
- **Ligne 536:** if (metadata?.expiresAt && new Date(metadata.expiresAt) < new Date()) {
- **Ligne 569:** expiresAt: input.expiresAt || null, // ✓ stocké dans metadata
- **Ligne 622:** defaultLanguage: z.enum(["en", "fr", "es", "de"]).optional(),
- **Ligne 764:** sslCertificateExpiry: new Date(Date.now() + 90 * 24 * 60 * 60 * 1000), // 90 days
- **Ligne 860:** description: Created email template: \${input.name} ,
- **Ligne 900:** description: Updated email template: \${template.name} ,
- **Ligne 926:** description: Deleted email template: \${template.name} ,
- **Ligne 1073:** expiresAt: new Date(Date.now() + 7 * 24 * 60 * 60 * 1000), // 7 jours
- **Ligne 1353:** description: "Updated login page branding",
- **Ligne 1362:** // ⚡ NAVIGATION MENU CONFIG
- **Ligne 1396:** description: "Updated navigation menu configuration",
- **Ligne 1430:** description: Updated email domain to \${input.customEmailDomain} ,
- **Ligne 1457:** description: "Verified email domain",

server/api/routers/timesheet.ts

- **Ligne 234:** message: "Start date must be before end date.",
- **Ligne 306:** date: new Date(cursor), // 🔥 FIX: Create a NEW Date object for each entry

- **Ligne 365:** description: expense.description,
- **Ligne 370:** expenseDate: start, // Use timesheet start date as expense date
- **Ligne 412:** date: z.date().optional(),
- **Ligne 698:** description: Work on \${new Date(entry.date).toISOString().slice(0, 10)} (\${entry.hours}h)\${entry.description ? ':' + entry.description : ''},
- **Ligne 709:** description: Expense: \${expense.title} - \${expense.description || ''},
- **Ligne 741:** dueDate: new Date(Date.now() + 30 * 24 * 60 * 60 * 1000),
- **Ligne 744:** notes: input.notes || Auto-generated from timesheet. Total hours: \${timesheet.totalHours}. Base amount: \${baseAmount}, Margin: \${marginAmount}, Expenses: \${totalExpenses}, Total: \${totalA`}
- **Ligne 751:** // ★ INCLUDE COMPLET POUR RETURN L'INVOICE COMPLÈTE
- **Ligne 806:** description: doc.description,
- **Ligne 883:** description: input.description,

server/api/routers/user.ts

- **Ligne 15:** // Permissions (tes clés existantes)
- **Ligne 29:** // Ownership helper: récupère toute la subtree
- **Ligne 35:** // On ne veut PAS ré-inclure rootUserId dans owned, donc on part de ses enfants
- **Ligne 53:** // - global -> tout voir
- **Ligne 105:** // - global -> tout voir
- **Ligne 210:** email: user.email,
- **Ligne 242:** email: z.string().email(),
- **Ligne 245:** // tu peux ajouter d'autres champs optionnels si besoin
- **Ligne 257:** email: input.email,
- **Ligne 264:** // Si password non fourni → on crée un token d'activation
- **Ligne 271:** expiresAt: new Date(Date.now() + 60 * 60 * 1000), // 1h
- **Ligne 286:** description: Created user \${newUser.name} (\${newUser.email}),
- **Ligne 347:** // - global → peut tout modifier
- **Ligne 348:** // - own → peut modifier self + subtree
- **Ligne 356:** email: z.string().email(),
- **Ligne 386:** email: input.email,
- **Ligne 478:** // - global only – à adapter selon ta stack auth
- **Ligne 505:** // TODO: génère un token d'impersonation/établit la session selon ton provider

server/api/routers/webhook.ts

- **Ligne 265:** hasNext: page < Math.ceil(total / pageSize),

server/helpers/contracts/contractPermissions.ts

- **Ligne 2:** * Helpers pour vérifier les permissions et l'accès aux contrats
- **Ligne 4:** * Ces helpers vérifient si un utilisateur a les droits nécessaires pour
- **Ligne 5:** * effectuer des actions sur un contrat (lecture, modification, upload de documents).
- **Ligne 11:** * Vérifie si un utilisateur est participant d'un contrat

- **Ligne 13:** * Un utilisateur est considéré comme participant si :
- **Ligne 14:** * - Il apparaît directement dans ContractParticipant (via userId)
- **Ligne 15:** * - Il est membre d'une company qui est participante (via companyId)
- **Ligne 18:** * @param contractId - ID du contrat
- **Ligne 19:** * @param userId - ID de l'utilisateur
- **Ligne 20:** * @returns true si l'utilisateur est participant, false sinon
- **Ligne 31:** // Vérifier si l'utilisateur est directement participant
- **Ligne 44:** // Vérifier si l'utilisateur est membre d'une company participante
- **Ligne 71:** * Vérifie si un utilisateur peut modifier un contrat
- **Ligne 73:** * Un utilisateur peut modifier un contrat si :
- **Ligne 74:** * - Il a la permission contract.update.global OU
- **Ligne 75:** * - Il a la permission contract.update.own ET est participant du contrat OU
- **Ligne 76:** * - Il est le créateur du contrat (createdBy) ET a contract.update.own
- **Ligne 78:** * De plus, certains statuts de contrat empêchent toute modification :
- **Ligne 79:** * - Les contrats "active" et "completed" ne peuvent plus être modifiés
- **Ligne 80:** * (sauf pour certaines actions spécifiques)
- **Ligne 83:** * @param contractId - ID du contrat
- **Ligne 84:** * @param userId - ID de l'utilisateur
- **Ligne 85:** * @param userPermissions - Permissions de l'utilisateur (clés des permissions)
- **Ligne 86:** * @returns true si l'utilisateur peut modifier, false sinon
- **Ligne 108:** // Récupérer le contrat pour vérifier le statut et le créateur
- **Ligne 121:** // Les contrats "completed" et "active" ne peuvent plus être modifiés
- **Ligne 122:** // (sauf pour uploader des documents signés, mais c'est géré ailleurs)
- **Ligne 124:** // Seul contract.update.global peut modifier ces contrats
- **Ligne 128:** // Permission "own" nécessite d'être participant ou créateur
- **Ligne 130:** // Vérifier si l'utilisateur est le créateur
- **Ligne 135:** // Vérifier si l'utilisateur est participant
- **Ligne 148:** * Vérifie si un utilisateur peut uploader des documents pour un contrat
- **Ligne 150:** * Un utilisateur peut uploader des documents si :
- **Ligne 151:** * - Il est participant du contrat (directement ou via company) ET
- **Ligne 152:** * - Le contrat n'est pas en statut "completed" ou "active"
- **Ligne 154:** * Exception: Les utilisateurs avec contract.update.global peuvent toujours uploader
- **Ligne 157:** * @param contractId - ID du contrat
- **Ligne 158:** * @param userId - ID de l'utilisateur
- **Ligne 159:** * @param userPermissions - Permissions de l'utilisateur (clés des permissions)
- **Ligne 160:** * @returns true si l'utilisateur peut uploader, false sinon
- **Ligne 182:** // Récupérer le contrat pour vérifier le statut
- **Ligne 194:** // Les contrats "completed" et "active" ne permettent plus l'upload
- **Ligne 195:** // (sauf pour contract.update.global, déjà vérifié plus haut)
- **Ligne 200:** // Vérifier si l'utilisateur est participant
- **Ligne 210:** * Vérifie si un utilisateur peut supprimer un document
- **Ligne 212:** * Un utilisateur peut supprimer un document si :
- **Ligne 213:** * - Il est l'uploader du document OU

- **Ligne 214:** * - Il a la permission contract.update.global
- **Ligne 216:** * De plus, le contrat ne doit pas être "completed" ou "active"
- **Ligne 217:** * (sauf pour contract.update.global)
- **Ligne 221:** * @param userId - ID de l'utilisateur
- **Ligne 222:** * @param userPermissions - Permissions de l'utilisateur (clés des permissions)
- **Ligne 223:** * @returns true si l'utilisateur peut supprimer, false sinon
- **Ligne 245:** // Récupérer le document avec le contrat associé
- **Ligne 261:** // Les contrats "completed" et "active" ne permettent plus la suppression
- **Ligne 267:** // Vérifier si l'utilisateur est l'uploader
- **Ligne 280:** * Vérifie si un utilisateur peut voir un contrat
- **Ligne 282:** * Un utilisateur peut voir un contrat si :
- **Ligne 283:** * - Il a la permission contract.read.global OU
- **Ligne 284:** * - Il a la permission contract.read.own ET est participant du contrat OU
- **Ligne 285:** * - Il est le créateur du contrat (createdBy)
- **Ligne 288:** * @param contractId - ID du contrat
- **Ligne 289:** * @param userId - ID de l'utilisateur
- **Ligne 290:** * @param userPermissions - Permissions de l'utilisateur (clés des permissions)
- **Ligne 291:** * @returns true si l'utilisateur peut voir, false sinon
- **Ligne 313:** // Récupérer le contrat pour vérifier le créateur
- **Ligne 325:** // Permission "own" nécessite d'être participant ou créateur
- **Ligne 327:** // Vérifier si l'utilisateur est le créateur
- **Ligne 332:** // Vérifier si l'utilisateur est participant

server/helpers/contracts/createMinimalParticipant.ts

- **Ligne 2:** * Helper pour créer des participants minimaux pour les contrats simplifiés
- **Ligne 4:** * Ce helper assure que les participants sont créés correctement avec les
- **Ligne 5:** * règles de validation appropriées (ex: approvers ne peuvent pas signer).
- **Ligne 22:** * Crée un participant minimal pour un contrat simplifié
- **Ligne 24:** * Règles de validation :
- **Ligne 25:** * - Soit userId, soit companyId doit être fourni (au moins un)
- **Ligne 26:** * - Par défaut: isActive=true, approved=false, requiresSignature=false
- **Ligne 27:** * - Les approvers ne peuvent JAMAIS avoir requiresSignature=true
- **Ligne 30:** * @param input - Données du participant
- **Ligne 31:** * @returns Participant créé
- **Ligne 32:** * @throws TRPCError si validation échoue
- **Ligne 56:** // Validation 1: Au moins userId ou companyId doit être fourni
- **Ligne 60:** message: "Au moins userId ou companyId doit être fourni pour créer un participant",
- **Ligne 64:** // Validation 2: Les approvers ne peuvent JAMAIS avoir requiresSignature=true
- **Ligne 68:** message: "Les approvers ne peuvent pas avoir requiresSignature=true. " +
- **Ligne 69:** "Les approvers approuvent, ils ne signent pas.",
- **Ligne 73:** // Créer le participant
- **Ligne 107:** message: "Échec de la création du participant",
- **Ligne 114:** * Crée automatiquement un participant "client" basé sur une company

- **Ligne 116:** * Raccourci pour créer un client primaire sans signature requise.
- **Ligne 119:** * @param contractId - ID du contrat
- **Ligne 120:** * @param companyId - ID de la company
- **Ligne 121:** * @returns Participant client créé
- **Ligne 142:** * Crée automatiquement un participant "contractor" basé sur un utilisateur
- **Ligne 144:** * Raccourci pour créer un contractor primaire avec signature requise.
- **Ligne 147:** * @param contractId - ID du contrat
- **Ligne 148:** * @param userId - ID de l'utilisateur contractor
- **Ligne 149:** * @returns Participant contractor créé
- **Ligne 170:** * Crée un participant "approver" (admin interne qui approuve le contrat)
- **Ligne 172:** * IMPORTANT: Les approvers n'ont jamais requiresSignature=true.
- **Ligne 173:** * Ils approuvent via le champ "approved", ils ne signent pas.
- **Ligne 176:** * @param contractId - ID du contrat
- **Ligne 177:** * @param userId - ID de l'utilisateur approver
- **Ligne 178:** * @returns Participant approver créé
- **Ligne 193:** requiresSignature: false, // ! CRITIQUE: Toujours false pour les approvers

server/helpers/contracts/generateContractTitle.ts

- **Ligne 2:** * Helper pour générer automatiquement un titre de contrat
- **Ligne 3:** * à partir d'un nom de fichier PDF
- **Ligne 5:** * Utilisé par le système simplifié de contrats MSA/SOW pour créer
- **Ligne 6:** * automatiquement des titres lisibles sans intervention de l'utilisateur.
- **Ligne 10:** * Génère un titre de contrat à partir d'un nom de fichier PDF
- **Ligne 12:** * Règles de transformation :
- **Ligne 14:** * - Remplacer underscores et tirets par des espaces
- **Ligne 15:** * - Capitaliser la première lettre de chaque mot
- **Ligne 16:** * - Limiter à 100 caractères
- **Ligne 17:** * - Retourner "Untitled Contract" si le résultat est vide
- **Ligne 19:** * @param fileName - Nom du fichier (ex: "msa_client_abc.pdf")
- **Ligne 20:** * @returns Titre formaté (ex: "Msa Client Abc")
- **Ligne 35:** // 2. Remplacer underscores, tirets, et caractères spéciaux par des espaces
- **Ligne 38:** // 3. Capitaliser première lettre de chaque mot
- **Ligne 48:** // 4. Limiter à 100 caractères
- **Ligne 53:** // 5. Retourner un titre par défaut si vide
- **Ligne 58:** * Génère un titre avec préfixe selon le type de contrat
- **Ligne 60:** * @param fileName - Nom du fichier
- **Ligne 61:** * @param type - Type de contrat ("msa" ou "sow")
- **Ligne 62:** * @returns Titre avec préfixe (ex: "[MSA] Client Abc")
- **Ligne 77:** // Limiter à 100 caractères en incluant le préfixe
- **Ligne 84:** * Génère un titre avec horodatage
- **Ligne 86:** * @param fileName - Nom du fichier
- **Ligne 87:** * @param addTimestamp - Ajouter un timestamp au titre
- **Ligne 88:** * @returns Titre avec timestamp (ex: "Client Abc - 2024-01-15")
- **Ligne 108:** // Limiter à 100 caractères

server/helpers/contracts/participantHelpers.ts

- **Ligne 2:** * Helpers pour la gestion des participants supplémentaires
- **Ligne 4:** * Ces helpers facilitent la création et la validation des participants
- **Ligne 5:** * lors de la création de contrats ou l'ajout manuel de participants.
- **Ligne 13:** * Crée plusieurs participants supplémentaires pour un contrat
- **Ligne 15:** * Cette fonction est utilisée lors de la création de contrats pour ajouter
- **Ligne 16:** * tous les participants supplémentaires en une seule transaction.
- **Ligne 19:** * @param contractId - ID du contrat
- **Ligne 20:** * @param participants - Tableau de participants à créer
- **Ligne 21:** * @returns Tableau des participants créés
- **Ligne 22:** * @throws TRPCError si validation échoue
- **Ligne 45:** // Validation: au moins userId ou companyId doit être fourni
- **Ligne 49:**
 - message: "Au moins userId ou companyId doit être fourni pour chaque participant",
- **Ligne 53:** // Vérifier si le participant n'existe pas déjà
- **Ligne 70:** // Créer le participant
- **Ligne 106:** message: "Échec de la création d'un participant supplémentaire",
- **Ligne 116:** * Vérifie si un participant peut être supprimé
- **Ligne 118:** * Les participants principaux (company_tenant, agency, contractor) ne peuvent
- **Ligne 119:** * pas être supprimés car ils sont essentiels au contrat.
- **Ligne 121:** * @param role - Rôle du participant
- **Ligne 122:** * @returns true si le participant peut être supprimé, false sinon
- **Ligne 134:** * Valide qu'un participant peut être ajouté à un contrat
- **Ligne 136:** * Vérifie que :
- **Ligne 137:** * - Le contrat existe et est dans un statut modifiable (draft ou pending)
- **Ligne 138:** * - Au moins userId ou companyId est fourni
- **Ligne 139:** * - L'utilisateur ou la company existent s'ils sont fournis
- **Ligne 142:** * @param contractId - ID du contrat
- **Ligne 143:** * @param userId - ID de l'utilisateur (optionnel)
- **Ligne 144:** * @param companyId - ID de la company (optionnel)
- **Ligne 145:** * @throws TRPCError si validation échoue
- **Ligne 156:** // Validation 1: Au moins userId ou companyId doit être fourni
- **Ligne 160:** message: "Au moins userId ou companyId doit être fourni",
- **Ligne 164:** // Validation 2: Le contrat existe et est modifiable
- **Ligne 180:** // Les contrats "completed" et "active" ne peuvent plus être modifiés
- **Ligne 184:** message: "Impossible d'ajouter des participants à un contrat complété ou actif",
- **Ligne 188:** // Validation 3: Vérifier que l'utilisateur existe (si fourni)
- **Ligne 203:** // Validation 4: Vérifier que la company existe (si fournie)
- **Ligne 220:** * Récupère la company associée à un utilisateur (si elle existe)
- **Ligne 222:** * Utile pour implémenter la fonctionnalité "lier la company du user"
- **Ligne 223:** * lors de la sélection d'un participant.
- **Ligne 226:** * @param userId - ID de l'utilisateur
- **Ligne 227:** * @returns Company associée ou null

- **Ligne 232:** * // Proposer de lier aussi la company
- **Ligne 240:** // Chercher une CompanyUser active pour cet utilisateur
- **Ligne 255:** createdAt: "desc", // Prendre la plus récente si plusieurs

server/helpers/contracts/simpleWorkflowTransitions.ts

- **Ligne 2:** * Helper pour gérer les transitions de workflow du système simplifié
- **Ligne 4:** * Ce helper définit et valide les transitions autorisées entre les
- **Ligne 5:** * différents statuts des contrats simplifiés.
- **Ligne 42:** // TRANSITIONS AUTORISÉES (Workflow Simplifié)
- **Ligne 46:** * Définit les transitions autorisées dans le workflow simplifié
- **Ligne 56:** * 8. * → cancelled (cancel - depuis n'importe quel statut)
- **Ligne 64:** description: "Soumettre le contrat pour validation admin",
- **Ligne 71:** description: "Approuver le contrat (admin)",
- **Ligne 78:** description: "Rejeter le contrat et le remettre en draft",
- **Ligne 85:** description: "Activer le contrat",
- **Ligne 92:** description: "Mettre le contrat en pause",
- **Ligne 99:** description: "Reprendre le contrat en pause",
- **Ligne 106:** description: "Terminer le contrat",
- **Ligne 115:** * Vérifie si une transition est autorisée
- **Ligne 117:** * @param from - Statut de départ
- **Ligne 118:** * @param to - Statut d'arrivée
- **Ligne 119:** * @param action - Action à effectuer
- **Ligne 120:** * @returns true si la transition est autorisée
- **Ligne 137:** * Valide une transition et lance une erreur si non autorisée
- **Ligne 139:** * @param from - Statut de départ
- **Ligne 140:** * @param to - Statut d'arrivée
- **Ligne 141:** * @param action - Action à effectuer
- **Ligne 142:** * @throws TRPCError si transition non autorisée
- **Ligne 159:** Transition non autorisée: \${from} → \${to} via \${action}. +
- **Ligne 160:** Actions disponibles depuis \${from}: \${availableActions} || "aucune"}. ,
- **Ligne 166:** * Récupère les transitions possibles depuis un statut donné
- **Ligne 168:** * @param currentStatus - Statut actuel du contrat
- **Ligne 169:** * @returns Liste des transitions possibles
- **Ligne 182:** * Récupère la transition correspondant à une action depuis un statut
- **Ligne 185:** * @param action - Action à effectuer
- **Ligne 186:** * @returns Transition trouvée ou undefined
- **Ligne 201:** // HELPERS DE STATUT
- **Ligne 205:** * Vérifie si un contrat est en draft
- **Ligne 207:** * @param contract - Contrat à vérifier
- **Ligne 208:** * @returns true si le contrat est en draft
- **Ligne 215:** * Vérifie si un contrat peut être supprimé
- **Ligne 217:** * Règle: seuls les contrats en draft peuvent être supprimés
- **Ligne 219:** * @param contract - Contrat à vérifier
- **Ligne 220:** * @returns true si le contrat peut être supprimé

- **Ligne 227:** * Vérifie si un contrat peut être modifié
- **Ligne 229:** * Règle: seuls les contrats en draft ou pending_admin_review peuvent être modifiés
- **Ligne 231:** * @param contract - Contrat à vérifier
- **Ligne 232:** * @returns true si le contrat peut être modifié
- **Ligne 239:** * Vérifie si un contrat est actif (peut générer des factures, payslips, etc.)
- **Ligne 241:** * @param contract - Contrat à vérifier
- **Ligne 242:** * @returns true si le contrat est actif
- **Ligne 249:** * Vérifie si un contrat est complété (toutes signatures collectées)
- **Ligne 251:** * @param contract - Contrat à vérifier
- **Ligne 252:** * @returns true si le contrat est complété
- **Ligne 263:** * Obtient la couleur du badge selon le statut (pour UI)
- **Ligne 265:** * @param status - Statut du contrat
- **Ligne 266:** * @returns Nom de couleur (Tailwind CSS)
- **Ligne 268:** * Obtient le label français du statut (pour UI)
- **Ligne 288:** * @param status - Statut du contrat
- **Ligne 289:** * @returns Label en français
- **Ligne 292:** * getStatusLabel("pending_admin_review") // "En attente de validation"
- **Ligne 297:** pending_admin_review: "En attente de validation",
- **Ligne 298:** completed: "Complété",
- **Ligne 300:** cancelled: "Annulé",
- **Ligne 302:** terminated: "Terminé",
- **Ligne 309:** * Obtient la description d'une action (pour UI)
- **Ligne 312:** * @returns Description en français
- **Ligne 315:** * getActionLabel("submit_for_review") // "Soumettre pour validation"
- **Ligne 319:** submit_for_review: "Soumettre pour validation",
- **Ligne 323:** pause: "Mettre en pause",

server/helpers/contracts/validateCompanies.ts

- **Ligne 2:** * Helper pour valider que les companies existent et sont actives
- **Ligne 4:** * Utilisé lors de la création d'un contrat NORM pour s'assurer que
- **Ligne 5:** * les companies (tenant et agency) existent et sont dans un état valide.
- **Ligne 12:** * Valide qu'une company existe et est active dans le tenant
- **Ligne 14:** * Règles de validation :
- **Ligne 15:** * - La company doit exister
- **Ligne 16:** * - La company doit appartenir au même tenant
- **Ligne 17:** * - La company doit être active (status="active")
- **Ligne 20:** * @param companyId - ID de la company à valider
- **Ligne 21:** * @param tenantId - ID du tenant (pour vérification de sécurité)
- **Ligne 22:** * @param companyType - Type de company ("tenant" ou "agency") pour messages d'erreur
- **Ligne 23:** * @returns Company validée
- **Ligne 24:** * @throws TRPCError si validation échoue
- **Ligne 35:** // 1. Récupérer la company

- **Ligne 59:** // 2. Vérifier que la company existe
- **Ligne 63:** message: Company \${companyType} introuvable. Vérifiez que l'ID est correct et que vous avez accès à cette company. ,
- **Ligne 67:** // 3. Vérifier que la company est active
- **Ligne 71:** message: La company \${companyType} "\${company.name}" est inactive (statut: \${company.status}) et ne peut pas être utilisée dans un contrat. ,
- **Ligne 79:** * Valide plusieurs companies en une seule opération
- **Ligne 81:** * Utile pour valider à la fois la company tenant et l'agency.
- **Ligne 84:** * @param companyTenantId - ID de la company tenant
- **Ligne 87:** * @returns Object contenant les deux companies validées
- **Ligne 88:** * @throws TRPCError si une validation échoue
- **Ligne 104:** // Valider les deux companies en parallèle
- **Ligne 110:** // Vérifier que ce ne sont pas la même company
- **Ligne 114:** message: "La company tenant et l'agency ne peuvent pas être la même company.",
- **Ligne 125:** * Récupère toutes les companies disponibles pour créer un contrat NORM
- **Ligne 127:** * Utile pour afficher une liste de companies dans un sélecteur UI.
- **Ligne 131:** * @param activeOnly - Ne retourner que les companies actives (par défaut: true)
- **Ligne 132:** * @returns Liste des companies disponibles

server/helpers/contracts/validateContractor.ts

- **Ligne 2:** * Helper pour valider qu'un utilisateur est bien un contractor
- **Ligne 4:** * Utilisé lors de la création d'un contrat NORM pour s'assurer que
- **Ligne 5:** * l'utilisateur sélectionné comme contractor a bien le rôle approprié.
- **Ligne 12:** * Valide qu'un utilisateur est bien un contractor actif du tenant
- **Ligne 14:** * Règles de validation :
- **Ligne 15:** * - L'utilisateur doit exister
- **Ligne 16:** * - L'utilisateur doit appartenir au même tenant
- **Ligne 17:** * - L'utilisateur doit avoir un rôle nommé "CONTRACTOR" (ou similaire)
- **Ligne 18:** * - L'utilisateur doit être actif (isActive=true)
- **Ligne 21:** * @param userId - ID de l'utilisateur à valider
- **Ligne 22:** * @param tenantId - ID du tenant (pour vérification de sécurité)
- **Ligne 23:** * @returns Utilisateur contractor validé avec son rôle
- **Ligne 24:** * @throws TRPCError si validation échoue
- **Ligne 35:** // 1. Récupérer l'utilisateur avec son rôle
- **Ligne 52:** // 2. Vérifier que l'utilisateur existe
- **Ligne 56:**
 - message: "Utilisateur introuvable. Vérifiez que l'ID est correct et que vous avez accès à cet utilisateur.",
- **Ligne 60:** // 3. Vérifier que l'utilisateur est actif
- **Ligne 64:** message: L'utilisateur "\${user.name || user.email}" est inactif et ne peut pas être assigné comme contractor. ,
- **Ligne 68:** // 4. Vérifier que l'utilisateur a le rôle CONTRACTOR
- **Ligne 78:** message: L'utilisateur "\${user.name || user.email}" n'a pas le rôle CONTRACTOR. +
- **Ligne 79:** Rôle actuel: \${user.role.displayName || user.role.name}. +

- **Ligne 80:** "Seuls les utilisateurs avec le rôle CONTRACTOR peuvent être assignés à un contrat NORM." ,
- **Ligne 88:** * Récupère tous les contractors disponibles pour créer un contrat NORM
- **Ligne 90:** * Utile pour afficher une liste de contractors dans un sélecteur UI.
- **Ligne 94:** * @param activeOnly - Ne retourner que les contractors actifs (par défaut: true)
- **Ligne 95:** * @returns Liste des contractors disponibles

server/helpers/contracts/validateParentMSA.ts

- **Ligne 2:** * Helper pour valider qu'un contrat parent est bien un MSA valide
- **Ligne 4:** * Utilisé lors de la création d'un SOW pour s'assurer que le parent
- **Ligne 5:** * existe, est un MSA, et est dans un état valide.
- **Ligne 12:** * Valide qu'un contrat parent est bien un MSA actif du même tenant
- **Ligne 14:** * Règles de validation :
- **Ligne 15:** * - Le contrat parent doit exister
- **Ligne 16:** * - Le contrat parent doit être du type "msa"
- **Ligne 17:** * - Le contrat parent doit appartenir au même tenant
- **Ligne 18:** * - Le contrat parent doit être dans un statut valide (pas cancelled)
- **Ligne 21:** * @param parentId - ID du contrat parent
- **Ligne 22:** * @param tenantId - ID du tenant (pour vérification de sécurité)
- **Ligne 23:** * @returns Contrat MSA parent avec ses participants
- **Ligne 24:** * @throws TRPCError si validation échoue
- **Ligne 35:** // 1. Récupérer le contrat parent
- **Ligne 63:** // 2. Vérifier que le parent existe
- **Ligne 67:** message: "MSA parent introuvable. Vérifiez que l'ID est correct et que vous avez accès à ce contrat.",
- **Ligne 71:** // 3. Vérifier que le parent est bien un MSA
- **Ligne 75:** message: Le contrat parent doit être un MSA. Type actuel: \${parent.type}. +
- **Ligne 76:** "Un SOW ne peut être lié qu'à un MSA, pas à un autre SOW.",
- **Ligne 80:** // 4. Vérifier que le MSA est dans un statut valide
- **Ligne 91:** message: Le MSA parent est en statut "\${parent.status}" et ne peut pas être utilisé. +
- **Ligne 96:** // 5. Optionnel: Avertir si le MSA parent est encore en draft
- **Ligne 108:** * Récupère tous les MSA disponibles pour créer un SOW
- **Ligne 110:** * Utile pour afficher une liste de MSA dans un sélecteur UI.
- **Ligne 114:** * @param activeOnly - Ne retourner que les MSA actifs (par défaut: false)
- **Ligne 115:** * @returns Liste des MSA disponibles
- **Ligne 133:** // Exclure seulement les cancelled et terminated

server/helpers/contracts/validatePaymentMethods.ts

- **Ligne 2:** * Helper pour valider que les PaymentMethods (UserBanks) existent
- **Ligne 4:** * Utilisé lors de la création d'un contrat NORM pour s'assurer que
- **Ligne 5:** * les méthodes de paiement sélectionnées existent et sont actives.
- **Ligne 12:** * Valide qu'une méthode de paiement existe et est active
- **Ligne 14:** * Règles de validation :

- **Ligne 15:** * - La méthode de paiement doit exister
- **Ligne 16:** * - Elle doit appartenir au même tenant
- **Ligne 17:** * - Elle doit être de type BANK_ACCOUNT
- **Ligne 18:** * - Elle doit être active (isActive=true)
- **Ligne 19:** * - Elle doit appartenir au contractor spécifié
- **Ligne 22:** * @param paymentMethodId - ID de la méthode de paiement à valider
- **Ligne 23:** * @param userId - ID du contractor propriétaire
- **Ligne 24:** * @param tenantId - ID du tenant (pour vérification de sécurité)
- **Ligne 25:** * @returns Méthode de paiement validée
- **Ligne 26:** * @throws TRPCError si validation échoue
- **Ligne 37:** // 1. Récupérer la méthode de paiement
- **Ligne 46:** // 2. Vérifier que la méthode de paiement existe
- **Ligne 50:** message: "Méthode de paiement introuvable. Vérifiez que l'ID est correct et qu'elle appartient au contractor.",
- **Ligne 54:** // 3. Vérifier que la méthode de paiement est active
- **Ligne 58:** message: "La méthode de paiement sélectionnée est inactive et ne peut pas être utilisée.",
- **Ligne 62:** // 4. Vérifier que c'est bien un compte bancaire
- **Ligne 66:** message: Seuls les comptes bancaires peuvent être utilisés pour les contrats NORM. Type actuel: \${paymentMethod.type}. ,
- **Ligne 74:** * Valide plusieurs méthodes de paiement (pour le mode Split)
- **Ligne 76:** * Utile pour valider un array de PaymentMethods.
- **Ligne 79:** * @param paymentMethodIds - Array d'IDs des méthodes de paiement
- **Ligne 80:** * @param userId - ID du contractor propriétaire
- **Ligne 82:** * @returns Array des méthodes de paiement validées
- **Ligne 83:** * @throws TRPCError si une validation échoue
- **Ligne 99:** // Vérifier qu'il y a au moins une méthode de paiement
- **Ligne 103:** message: "Au moins une méthode de paiement doit être fournie pour le mode Split.",
- **Ligne 107:** // Valider toutes les méthodes de paiement en parallèle
- **Ligne 114:** // Vérifier qu'il n'y a pas de doublons
- **Ligne 119:** message: "Les méthodes de paiement en double ne sont pas autorisées.",
- **Ligne 127:** * Récupère toutes les méthodes de paiement disponibles pour un contractor
- **Ligne 129:** * Utile pour afficher une liste de comptes bancaires dans un sélecteur UI.
- **Ligne 134:** * @param activeOnly - Ne retourner que les méthodes actives (par défaut: true)
- **Ligne 135:** * @returns Liste des méthodes de paiement disponibles

server/rbac/permissions.ts

- **Ligne 2:** * Scopes de permissions pour le contexte
- **Ligne 5:** GLOBAL = "global", // Accès à toutes les ressources du tenant
- **Ligne 6:** OWN = "own", // Accès uniquement à ses propres ressources
- **Ligne 7:** TENANT = "tenant", // Accès au niveau tenant (équivalent à global pour certaines ressources)
- **Ligne 8:** PAGE = "page",

- **Ligne 12:** * Ressources disponibles dans la plateforme
- **Ligne 53:** EMAIL = "email",
- **Ligne 125:** // Liste et recherche
- **Ligne 129:** // Actions spécifiques
- **Ligne 184:** * Type pour une permission complète
- **Ligne 197:** * Catégories de permissions pour l'UI
- **Ligne 201:** BUSINESS = "Gestion métier",
- **Ligne 215:** * Construit une clé de permission
- **Ligne 226:** * Parse une clé de permission
- **Ligne 244:** * Crée un objet Permission
- **Ligne 266:** // PERMISSIONS COMPLÈTES VOOR PLATEFORME DEEL-LIKE
- **Ligne 278:** "Voir son dashboard",
- **Ligne 279:** "Accéder à son tableau de bord personnel",
- **Ligne 289:** "Voir son dashboard",
- **Ligne 290:** "Accéder à son tableau de bord personnel",
- **Ligne 297:** "Voir tous les dashboards",
- **Ligne 298:** "Accéder aux statistiques globales du tenant",
- **Ligne 309:** "Voir son profil",
- **Ligne 310:** "Consulter et gérer son propre profil utilisateur",
- **Ligne 317:** "Modifier son profil",
- **Ligne 318:** "Mettre à jour ses informations personnelles",
- **Ligne 325:** "Voir les détails des utilisateurs",
- **Ligne 326:** "Consulter les profils et informations détaillées de tous les utilisateurs",
- **Ligne 333:** "Voir tous les utilisateurs",
- **Ligne 334:** "Lister et rechercher tous les utilisateurs du tenant",
- **Ligne 341:** "Créer des utilisateurs",
- **Ligne 342:** "Ajouter de nouveaux utilisateurs",
- **Ligne 349:** "Modifier les utilisateurs",
- **Ligne 350:** "Mettre à jour les informations des utilisateurs",
- **Ligne 357:** "Supprimer des utilisateurs",
- **Ligne 358:** "Supprimer des comptes utilisateurs",
- **Ligne 365:** "Activer des utilisateurs",
- **Ligne 366:** "Activer ou désactiver des comptes",
- **Ligne 373:** "Se connecter en tant qu'utilisateur",
- **Ligne 374:** "Impersonner un autre utilisateur",
- **Ligne 385:** "Voir les rôles",
- **Ligne 386:** "Lister tous les rôles",
- **Ligne 393:** "Créer des rôles",
- **Ligne 394:** "Créer de nouveaux rôles personnalisés",
- **Ligne 401:** "Modifier des rôles",
- **Ligne 402:** "Modifier les rôles existants",
- **Ligne 409:** "Supprimer des rôles",
- **Ligne 410:** "Supprimer des rôles (sauf système)",
- **Ligne 417:** "Voir ses propres rôles",

- **Ligne 418:** "Lire uniquement les rôles que l'utilisateur a créés",
- **Ligne 425:** "Créer ses propres rôles",
- **Ligne 426:** "Créer un rôle qui sera marqué comme appartenant à l'utilisateur",
- **Ligne 433:** "Modifier ses propres rôles",
- **Ligne 434:** "Modifier uniquement les rôles que l'utilisateur a créés",
- **Ligne 441:** "Supprimer ses propres rôles",
- **Ligne 442:** "Supprimer uniquement les rôles créés par l'utilisateur (hors rôles système)",
- **Ligne 453:** "Voir les permissions",
- **Ligne 454:** "Lister toutes les permissions disponibles",
- **Ligne 462:** "Attribuer des permissions aux rôles",
- **Ligne 469:** "Créer des permissions",
- **Ligne 470:** "Créer de nouvelles permissions personnalisées",
- **Ligne 483:** "Voir toutes les entreprises",
- **Ligne 484:** "Lister toutes les entreprises du tenant",
- **Ligne 491:** "Voir ses entreprises",
- **Ligne 492:** "Lister uniquement les entreprises appartenant à l'utilisateur",
- **Ligne 501:** "Créer des entreprises (global)",
- **Ligne 502:** "Créer des entreprises appartenant au tenant",
- **Ligne 509:** "Créer ses propres entreprises",
- **Ligne 510:** "Créer des entreprises appartenant à l'utilisateur",
- **Ligne 519:** "Modifier toutes les entreprises",
- **Ligne 520:** "Mettre à jour n'importe quelle entreprise du tenant",
- **Ligne 527:** "Modifier ses entreprises",
- **Ligne 528:** "Mettre à jour uniquement les entreprises appartenant à l'utilisateur",
- **Ligne 537:** "Supprimer des entreprises (global)",
- **Ligne 538:** "Supprimer n'importe quelle entreprise du tenant",
- **Ligne 545:** "Supprimer ses entreprises",
- **Ligne 546:** "Supprimer uniquement les entreprises appartenant à l'utilisateur",
- **Ligne 558:** "Voir ses contrats",
- **Ligne 566:** "Assigner les participants à un contrat",
- **Ligne 567:** "Permet d'ajouter un admin, un approver ou toute autre personne sur un contrat",
- **Ligne 574:** "Voir tous les contrats",
- **Ligne 575:** "Lister et rechercher tous les contrats",
- **Ligne 582:** "Créer des contrats",
- **Ligne 583:** "Créer de nouveaux contrats",
- **Ligne 590:** "Modifier ses contrats",
- **Ligne 591:** "Mettre à jour ses propres contrats (draft uniquement)",
- **Ligne 598:** "Modifier tous les contrats",
- **Ligne 599:** "Mettre à jour n'importe quel contrat",
- **Ligne 606:** "Supprimer des contrats",
- **Ligne 607:** "Supprimer des contrats (draft uniquement)",
- **Ligne 614:** "Envoyer des contrats",
- **Ligne 615:** "Envoyer des contrats pour signature",

- **Ligne 623:** "Signer électroniquement ses contrats",
- **Ligne 630:** "Approuver des contrats",
- **Ligne 631:** "Approuver et activer des contrats",
- **Ligne 638:** "Annuler des contrats",
- **Ligne 639:** "Annuler ou résilier des contrats",
- **Ligne 646:** "Exporter des contrats",
- **Ligne 647:** "Exporter les données de contrats",
- **Ligne 654:** "Lier un SOW à un MSA",
- **Ligne 655:** "Créer ou rattacher un SOW à un contrat MSA",
- **Ligne 667:** "Voir tous les MSA",
- **Ligne 668:** "Lister tous les Master Service Agreements",
- **Ligne 675:** "Voir un MSA",
- **Ligne 676:** "Consulter les Master Service Agreements",
- **Ligne 683:** "Créer un MSA",
- **Ligne 684:** "Créer un Master Service Agreement",
- **Ligne 691:** "Modifier un MSA",
- **Ligne 692:** "Mettre à jour un Master Service Agreement",
- **Ligne 699:** "Supprimer un MSA",
- **Ligne 700:** "Supprimer un Master Service Agreement (non signé)",
- **Ligne 707:** "Envoyer un MSA",
- **Ligne 708:** "Envoyer un Master Service Agreement pour signature",
- **Ligne 716:** "Signer un Master Service Agreement",
- **Ligne 723:** "Approuver un MSA",
- **Ligne 724:** "Approuver un Master Service Agreement",
- **Ligne 731:** "Annuler un MSA",
- **Ligne 732:** "Annuler ou résilier un Master Service Agreement",
- **Ligne 739:** "Exporter les MSA",
- **Ligne 740:** "Exporter la liste des Master Service Agreements",
- **Ligne 750:** "Voir tous les SOW",
- **Ligne 751:** "Lister tous les Statements of Work",
- **Ligne 759:** "Voir ses SOW",
- **Ligne 760:** "Consulter les SOW liés à mes contrats",
- **Ligne 767:** "Voir tous les SOW",
- **Ligne 776:** "Créer un SOW",
- **Ligne 777:** "Créer un Statement of Work rattaché à un MSA",
- **Ligne 785:** "Modifier ses SOW",
- **Ligne 786:** "Mettre à jour ses propres SOW (draft uniquement)",
- **Ligne 793:** "Modifier tous les SOW",
- **Ligne 794:** "Mettre à jour n'importe quel Statement of Work",
- **Ligne 802:** "Supprimer un SOW",
- **Ligne 803:** "Supprimer un Statement of Work (non signé)",
- **Ligne 811:** "Envoyer un SOW",
- **Ligne 812:** "Envoyer un Statement of Work pour signature",
- **Ligne 829:** "Approuver un SOW",

- **Ligne 830:** "Approuver un Statement of Work",
- **Ligne 838:** "Annuler un SOW",
- **Ligne 839:** "Annuler ou résilier un Statement of Work",
- **Ligne 847:** "Exporter les SOW",
- **Ligne 848:** "Exporter la liste des Statements of Work",
- **Ligne 859:** "Voir ses documents de contrat",
- **Ligne 860:** "Consulter les documents de ses contrats",
- **Ligne 867:** "Voir tous les documents de contrat",
- **Ligne 868:** "Consulter tous les documents",
- **Ligne 876:** "Ajouter des documents à ses contrats",
- **Ligne 884:** "Ajouter des documents à n'importe quel contrat",
- **Ligne 891:** "Supprimer des documents",
- **Ligne 892:** "Supprimer des documents de contrat",
- **Ligne 903:** "Voir ses factures",
- **Ligne 911:** "Créer ses factures",
- **Ligne 912:** "Créer ses propres factures (contractors)",
- **Ligne 919:** "Voir toutes les factures",
- **Ligne 920:** "Lister et rechercher toutes les factures",
- **Ligne 927:** "Créer des factures",
- **Ligne 928:** "Créer des factures pour n'importe quel contrat",
- **Ligne 935:** "Modifier ses factures",
- **Ligne 936:** "Mettre à jour ses factures (draft uniquement)",
- **Ligne 943:** "Modifier toutes les factures",
- **Ligne 944:** "Mettre à jour n'importe quelle facture",
- **Ligne 951:** "Supprimer des factures",
- **Ligne 952:** "Supprimer des factures (draft uniquement)",
- **Ligne 959:** "Envoyer des factures",
- **Ligne 960:** "Envoyer des factures aux clients",
- **Ligne 967:** "Approuver des factures",
- **Ligne 968:** "Valider des factures avant envoi",
- **Ligne 975:** "Payer ses factures",
- **Ligne 976:** "Marquer ses propres factures comme payées (pour les agences)",
- **Ligne 983:** "Marquer comme payée",
- **Ligne 984:** "Marquer des factures comme payées",
- **Ligne 991:** "Exporter des factures",
- **Ligne 992:** "Exporter les données de factures",
- **Ligne 999:** "Confirmer la marge de ses factures",
- **Ligne 1000:** "Confirmer et valider la marge sur ses propres factures (agences)",
- **Ligne 1007:** "Réviser les factures",
- **Ligne 1008:** "Réviser et mettre en révision les factures",
- **Ligne 1015:** "Rejeter les factures",
- **Ligne 1016:** "Rejeter des factures avec raison",
- **Ligne 1023:** "Modifier les montants et marges",
- **Ligne 1024:** "Modifier les montants et marges des factures (admin)",

- **Ligne 1031:** "Marquer ses factures comme payées",
- **Ligne 1032:** "Marquer comme payées les factures dont on est le destinataire (agences)",
- **Ligne 1039:** "Confirmer la réception du paiement",
- **Ligne 1040:** "Confirmer que le paiement a été reçu avec le montant exact (admin)",
- **Ligne 1051:** "Voir ses paiements",
- **Ligne 1059:** "Voir tous les paiements",
- **Ligne 1060:** "Lister tous les paiements",
- **Ligne 1067:** "Créer des paiements",
- **Ligne 1068:** "Créer de nouveaux paiements",
- **Ligne 1075:** "Modifier les paiements",
- **Ligne 1076:** "Mettre à jour des paiements (y compris confirmation)",
- **Ligne 1083:** "Supprimer des paiements",
- **Ligne 1084:** "Supprimer des paiements non complétés",
- **Ligne 1091:** "Voir les paiements",
- **Ligne 1092:** "Consulter les paiements du tenant",
- **Ligne 1099:** "Traiter des paiements",
- **Ligne 1100:** "Traiter et finaliser des paiements",
- **Ligne 1108:** "Émettre des remboursements",
- **Ligne 1115:** "Exporter des paiements",
- **Ligne 1116:** "Exporter les données de paiements",
- **Ligne 1127:** "Voir ses dépenses",
- **Ligne 1128:** "Consulter ses propres dépenses",
- **Ligne 1135:** "Créer des dépenses",
- **Ligne 1136:** "Soumettre des notes de frais",
- **Ligne 1143:** "Modifier ses dépenses",
- **Ligne 1144:** "Mettre à jour ses dépenses (draft/rejected)",
- **Ligne 1151:** "Supprimer ses dépenses",
- **Ligne 1152:** "Supprimer ses dépenses (draft uniquement)",
- **Ligne 1159:** "Soumettre ses dépenses",
- **Ligne 1160:** "Soumettre des dépenses pour approbation",
- **Ligne 1167:** "Voir toutes les dépenses",
- **Ligne 1168:** "Lister toutes les dépenses",
- **Ligne 1175:** "Approuver toutes les dépenses",
- **Ligne 1176:** "Approuver n'importe quelle dépense",
- **Ligne 1183:** "Rejeter des dépenses",
- **Ligne 1184:** "Rejeter des demandes de dépenses",
- **Ligne 1191:** "Marquer comme payée",
- **Ligne 1192:** "Marquer des dépenses comme remboursées",
- **Ligne 1203:** "Voir ses feuilles de temps",
- **Ligne 1211:** "Créer des feuilles de temps",
- **Ligne 1212:** "Créer de nouvelles timesheets",
- **Ligne 1219:** "Modifier ses feuilles de temps",
- **Ligne 1220:** "Mettre à jour ses timesheets (draft uniquement)",
- **Ligne 1227:** "Supprimer ses feuilles de temps",

- **Ligne 1228:** "Supprimer ses timesheets (draft uniquement)",
- **Ligne 1235:** "Soumettre ses feuilles de temps",
- **Ligne 1236:** "Soumettre des timesheets pour approbation",
- **Ligne 1243:** "Voir toutes les feuilles de temps",
- **Ligne 1244:** "Lister toutes les timesheets",
- **Ligne 1251:** "Réviser les feuilles de temps",
- **Ligne 1252:** "Marquer les timesheets comme en cours de révision",
- **Ligne 1259:** "Approuver toutes les feuilles de temps",
- **Ligne 1267:** "Rejeter des feuilles de temps",
- **Ligne 1268:** "Rejeter des timesheets",
- **Ligne 1275:** "Modifier toutes les feuilles de temps",
- **Ligne 1276:** "Modifier n'importe quelle timesheet (y compris les montants)",
- **Ligne 1283:** "Voir les marges des feuilles de temps",
- **Ligne 1284:** "Consulter les détails de marge et la répartition complète des montants dans les timesheets",
- **Ligne 1295:** "Voir ses bulletins de paie",
- **Ligne 1304:** "Voir tous les bulletins de paie",
- **Ligne 1305:** "Consulter les bulletins de tous les utilisateurs",
- **Ligne 1313:** "Lister les bulletins de paie",
- **Ligne 1314:** "Accéder à la liste complète des bulletins",
- **Ligne 1322:** "Générer des bulletins de paie",
- **Ligne 1323:** "Créer de nouveaux bulletins de paie pour un utilisateur",
- **Ligne 1331:** "Modifier les bulletins de paie",
- **Ligne 1332:** "Mettre à jour les bulletins existants",
- **Ligne 1340:** "Supprimer les bulletins de paie",
- **Ligne 1341:** "Supprimer les bulletins de paie existants",
- **Ligne 1349:** "Envoyer des bulletins de paie",
- **Ligne 1350:** "Envoyer les bulletins par email aux utilisateurs",
- **Ligne 1358:** "Exporter les bulletins de paie",
- **Ligne 1359:** "Télécharger ou exporter un bulletin de paie en PDF/CSV",
- **Ligne 1369:** "Voir ses virements",
- **Ligne 1378:** "Créer des demandes de virement",
- **Ligne 1385:** "Voir tous les virements",
- **Ligne 1386:** "Lister tous les virements",
- **Ligne 1393:** "Traiter des virements",
- **Ligne 1394:** "Traiter et finaliser des virements",
- **Ligne 1398:** // ★ AJOUTÉS → nouvelles permissions admin
- **Ligne 1403:** "Voir tous les détails des virements",
- **Ligne 1404:** "Consulter les informations détaillées de toutes les remittances du tenant",
- **Ligne 1412:** "Mettre à jour un virement",
- **Ligne 1413:** "Modifier une remittance : notes, statut, dates, etc.",
- **Ligne 1421:** "Supprimer un virement",
- **Ligne 1422:** "Supprimer une remittance existante",
- **Ligne 1434:** "Voir ses parrainages",

- **Ligne 1442:** "Créer des parrainages",
- **Ligne 1450:** "Voir tous les parrainages",
- **Ligne 1451:** "Lister tous les parrainages",
- **Ligne 1458:** "Approuver des parrainages",
- **Ligne 1459:** "Valider des parrainages",
- **Ligne 1466:** "Payer les récompenses",
- **Ligne 1467:** "Payer les récompenses de parrainage",
- **Ligne 1478:** "Voir ses tâches",
- **Ligne 1479:** "Consulter ses propres tâches",
- **Ligne 1486:** "Voir toutes les tâches",
- **Ligne 1487:** "Consulter toutes les tâches du tenant",
- **Ligne 1494:** "Créer des tâches",
- **Ligne 1495:** "Créer de nouvelles tâches",
- **Ligne 1502:** "Modifier ses tâches",
- **Ligne 1503:** "Mettre à jour ses tâches",
- **Ligne 1510:** "Modifier toutes les tâches",
- **Ligne 1511:** "Mettre à jour n'importe quelle tâche",
- **Ligne 1518:** "Supprimer des tâches",
- **Ligne 1519:** "Supprimer des tâches",
- **Ligne 1526:** "Assigner des tâches",
- **Ligne 1527:** "Assigner des tâches à des utilisateurs",
- **Ligne 1538:** "Voir les prospects",
- **Ligne 1539:** "Lister tous les prospects",
- **Ligne 1546:** "Créer des prospects",
- **Ligne 1547:** "Ajouter de nouveaux prospects",
- **Ligne 1554:** "Modifier les prospects",
- **Ligne 1555:** "Mettre à jour les prospects",
- **Ligne 1562:** "Supprimer des prospects",
- **Ligne 1563:** "Supprimer des prospects",
- **Ligne 1571:** "Assigner des prospects à des commerciaux",
- **Ligne 1582:** "Voir ses documents",
- **Ligne 1590:** "Voir tous les documents",
- **Ligne 1591:** "Consulter les documents de toutes les entités du tenant",
- **Ligne 1599:** "Ajouter de nouveaux documents pour ses propres entités",
- **Ligne 1607:** "Ajouter des documents pour n'importe quelle entité",
- **Ligne 1614:** "Mettre à jour tous documents",
- **Ligne 1615:** "Mettre à jour (nouvelle version) n'importe quel document du tenant",
- **Ligne 1622:** "Supprimer ses documents",
- **Ligne 1623:** "Supprimer ses propres documents",
- **Ligne 1630:** "Supprimer tous documents",
- **Ligne 1638:** "Mettre à jour ses documents",
- **Ligne 1639:** "Mettre à jour (nouvelle version) ses propres documents",
- **Ligne 1646:** "Lister tous les documents",
- **Ligne 1647:** "Permet de voir la liste de tous les documents du tenant",

- **Ligne 1658:** "Voir les templates d'onboarding",
- **Ligne 1666:** "Créer des templates d'onboarding",
- **Ligne 1667:** "Créer de nouveaux templates",
- **Ligne 1674:** "Modifier les templates d'onboarding",
- **Ligne 1675:** "Mettre à jour les templates",
- **Ligne 1682:** "Supprimer des templates d'onboarding",
- **Ligne 1683:** "Supprimer des templates",
- **Ligne 1690:** "Voir ses réponses d'onboarding",
- **Ligne 1691:** "Consulter ses réponses",
- **Ligne 1698:** "Soumettre ses réponses",
- **Ligne 1699:** "Soumettre les réponses d'onboarding",
- **Ligne 1706:** "Voir toutes les réponses d'onboarding",
- **Ligne 1707:** "Lister toutes les réponses",
- **Ligne 1714:** "Reviewer les réponses",
- **Ligne 1715:** "Reviewer et approuver les réponses",
- **Ligne 1726:** "Ajouter des commentaires",
- **Ligne 1727:** "Commenter sur les ressources accessibles",
- **Ligne 1734:** "Modifier ses commentaires",
- **Ligne 1735:** "Modifier ses propres commentaires",
- **Ligne 1742:** "Supprimer ses commentaires",
- **Ligne 1743:** "Supprimer ses propres commentaires",
- **Ligne 1750:** "Supprimer tous commentaires",
- **Ligne 1762:** "Voir les workflows d'approbation",
- **Ligne 1770:** "Créer des workflows",
- **Ligne 1771:** "Créer des workflows d'approbation",
- **Ligne 1778:** "Modifier les workflows",
- **Ligne 1779:** "Mettre à jour les workflows",
- **Ligne 1789:** "Voir toutes les banques",
- **Ligne 1790:** "Lister toutes les banques du tenant",
- **Ligne 1797:** "Voir mes banques",
- **Ligne 1798:** "Lister uniquement les banques créées par l'utilisateur",
- **Ligne 1806:** "Créer des banques (global)",
- **Ligne 1807:** "Ajouter des banques visibles par tout le tenant",
- **Ligne 1814:** "Créer des banques (own)",
- **Ligne 1815:** "Ajouter des banques personnelles",
- **Ligne 1823:** "Modifier des banques (global)",
- **Ligne 1824:** "Mettre à jour toutes les banques",
- **Ligne 1831:** "Modifier mes banques",
- **Ligne 1832:** "Mettre à jour uniquement les banques créées par l'utilisateur",
- **Ligne 1840:** "Supprimer des banques (global)",
- **Ligne 1848:** "Supprimer mes banques",
- **Ligne 1849:** "Supprimer uniquement les banques créées par l'utilisateur",
- **Ligne 1860:** "Voir les webhooks",
- **Ligne 1861:** "Lister tous les webhooks",

- **Ligne 1868:** "Créer des webhooks",
- **Ligne 1876:** "Modifier les webhooks",
- **Ligne 1877:** "Mettre à jour les webhooks",
- **Ligne 1884:** "Supprimer des webhooks",
- **Ligne 1885:** "Supprimer des webhooks",
- **Ligne 1896:** "Voir ses clés API",
- **Ligne 1897:** "Lister ses propres clés API",
- **Ligne 1904:** "Créer des clés API",
- **Ligne 1905:** "Générer de nouvelles clés API",
- **Ligne 1912:** "Supprimer ses clés API",
- **Ligne 1913:** "Révoquer ses clés API",
- **Ligne 1920:** "Voir toutes les clés API",
- **Ligne 1921:** "Lister toutes les clés API du tenant",
- **Ligne 1932:** "Voir les informations du tenant",
- **Ligne 1933:** "Consulter les informations de l'organisation",
- **Ligne 1940:** "Modifier le tenant",
- **Ligne 1941:** "Mettre à jour les paramètres de l'organisation",
- **Ligne 1949:** "Gérer les configurations avancées",
- **Ligne 1960:** "Voir les paramètres",
- **Ligne 1961:** "Consulter les paramètres système",
- **Ligne 1968:** "Modifier les paramètres",
- **Ligne 1969:** "Mettre à jour les paramètres système",
- **Ligne 1980:** "Voir les champs personnalisés",
- **Ligne 1981:** "Lister les champs personnalisés",
- **Ligne 1988:** "Créer des champs personnalisés",
- **Ligne 1989:** "Ajouter de nouveaux champs",
- **Ligne 1996:** "Modifier les champs personnalisés",
- **Ligne 1997:** "Mettre à jour les champs",
- **Ligne 2004:** "Supprimer des champs personnalisés",
- **Ligne 2005:** "Supprimer des champs",
- **Ligne 2016:** "Voir les tags",
- **Ligne 2017:** "Lister tous les tags",
- **Ligne 2024:** "Créer des tags",
- **Ligne 2025:** "Créer de nouveaux tags",
- **Ligne 2032:** "Modifier les tags",
- **Ligne 2033:** "Mettre à jour les tags",
- **Ligne 2040:** "Supprimer des tags",
- **Ligne 2041:** "Supprimer des tags",
- **Ligne 2052:** "Voir ses rapports",
- **Ligne 2060:** "Voir tous les rapports",
- **Ligne 2061:** "Accéder à tous les rapports",
- **Ligne 2068:** "Créer des rapports",
- **Ligne 2069:** "Générer de nouveaux rapports",
- **Ligne 2076:** "Exporter des rapports",

- **Ligne 2077:** "Exporter les rapports en PDF/Excel",
- **Ligne 2088:** "Voir les participants de contrat",
- **Ligne 2089:** "Lister tous les participants des contrats",
- **Ligne 2096:** "Ajouter un participant",
- **Ligne 2097:** "Associer un utilisateur à un contrat",
- **Ligne 2104:** "Modifier les participants",
- **Ligne 2105:** "Mettre à jour le rôle ou les informations d'un participant",
- **Ligne 2113:** "Retirer un utilisateur d'un contrat",
- **Ligne 2120:** "Voir sa participation aux contrats",
- **Ligne 2121:** "Voir sa propre relation avec les contrats",
- **Ligne 2133:** "Voir les logs d'audit",
- **Ligne 2134:** "Consulter l'historique des actions",
- **Ligne 2141:** "Exporter les logs",
- **Ligne 2142:** "Exporter les logs d'audit",
- **Ligne 2153:** "Envoyer des emails",
- **Ligne 2154:** "Envoyer des emails aux utilisateurs",
- **Ligne 2161:** "Voir l'historique des emails",
- **Ligne 2169:** "Créer des templates d'email",
- **Ligne 2170:** "Créer et gérer des templates d'email",
- **Ligne 2177:** createPermission(Resource.DASHBOARD, Action.ACCESS, PermissionScope.PAGE, "Accéder à Agency Invoices"),
- **Ligne 2178:**
createPermission(Resource.AGENCY_INVOICE, Action.ACCESS, PermissionScope.PAGE, "Accéder à Agency Invoices"),
- **Ligne 2179:** createPermission(Resource.CONSTRUCTION, Action.ACCESS, PermissionScope.PAGE, "Accéder à Construction"),
- **Ligne 2180:** createPermission(Resource.CONTRACT, Action.ACCESS, PermissionScope.PAGE, "Accéder à Contracts"),
- **Ligne 2181:** createPermission(Resource.EXPENSE, Action.ACCESS, PermissionScope.PAGE, "Accéder à Expenses"),
- **Ligne 2182:** createPermission(Resource.INVOICE, Action.ACCESS, PermissionScope.PAGE, "Accéder à Invoices"),
- **Ligne 2183:**
createPermission(Resource.LEAD, Action.ACCESS, PermissionScope.PAGE, "Accéder à Leads"),
- **Ligne 2184:** createPermission(Resource.ONBOARDING, Action.ACCESS, PermissionScope.PAGE, "Accéder à Onboarding"),
- **Ligne 2185:** createPermission(Resource.PAYMENT, Action.ACCESS, PermissionScope.PAGE, "Accéder à Payments"),
- **Ligne 2186:** createPermission(Resource.PAYSLIP, Action.ACCESS, PermissionScope.PAGE, "Accéder à Payslips"),
- **Ligne 2187:** createPermission(Resource.PROFILE, Action.ACCESS, PermissionScope.PAGE, "Accéder à Profile"),
- **Ligne 2188:** createPermission(Resource.REFERRAL, Action.ACCESS, PermissionScope.PAGE, "Accéder à Referrals"),

- **Ligne 2189:**

```
createPermission(Resource.REPORT, Action.ACCESS, PermissionScope.PAGE, "Accéder à Reports"),
```

- **Ligne 2190:** `createPermission(Resource.SETTINGS, Action.ACCESS, PermissionScope.PAGE, "Accéder à Settings")`,

- **Ligne 2191:** `createPermission(Resource.SUPERADMIN, Action.ACCESS, PermissionScope.PAGE, "Accéder à Superadmin")`,

- **Ligne 2192:**

```
createPermission(Resource.TASK, Action.ACCESS, PermissionScope.PAGE, "Accéder à Tasks") ,
```

- **Ligne 2193:** `createPermission(Resource.TIMESHEET, Action.ACCESS, PermissionScope.PAGE, "Accéder à Timesheets")`,

- **Ligne 2194:**

```
createPermission(Resource.USER, Action.ACCESS, PermissionScope.PAGE, "Accéder à Users") ,
```

- **Ligne 2195:** `createPermission(Resource.ONBOARDING_TEMPLATE, Action.ACCESS, PermissionScope.PAGE, "Accéder aux Templates")`,

- **Ligne 2196:** `createPermission(Resource.PAYSLIP, Action.ACCESS, PermissionScope.PAGE, "Accéder aux Payslips de Payments")`,

- **Ligne 2197:** `createPermission(Resource.REMITTANCE, Action.ACCESS, PermissionScope.PAGE, "Accéder aux Remittances")`,

- **Ligne 2198:** `createPermission(Resource.ACTIVITY_LOG, Action.ACCESS, PermissionScope.PAGE, "Accéder à Activity Logs")`,

- **Ligne 2199:**

```
createPermission(Resource.EMAIL, Action.ACCESS, PermissionScope.PAGE, "Accéder à l'envoi d'emails") ,
```

- **Ligne 2200:** `createPermission(Resource.EMAIL_LOG, Action.ACCESS, PermissionScope.PAGE, "Accéder à Email Logs")`,

- **Ligne 2201:** `createPermission(Resource.SMS_LOG, Action.ACCESS, PermissionScope.PAGE, "Accéder à SMS Logs")`,

- **Ligne 2202:** `createPermission(Resource.USER_ACTIVITY, Action.ACCESS, PermissionScope.PAGE, "Accéder à User Activity")`,

- **Ligne 2203:** `createPermission(Resource.BANK, Action.ACCESS, PermissionScope.PAGE, "Accéder aux Banks")`,

- **Ligne 2204:** `createPermission(Resource.BRANDING, Action.ACCESS, PermissionScope.PAGE, "Accéder à Branding")`,

- **Ligne 2205:**

```
createPermission(Resource.LOGIN, Action.ACCESS, PermissionScope.PAGE, "Accéder à Branding Login") ,
```

- **Ligne 2206:** `createPermission(Resource.COMPANY, Action.ACCESS, PermissionScope.PAGE, "Accéder aux Companies")`,

- **Ligne 2207:** `createPermission(Resource.COUNTRY, Action.ACCESS, PermissionScope.PAGE, "Accéder aux Countries")`,

- **Ligne 2208:** `createPermission(Resource.CURRENCY, Action.ACCESS, PermissionScope.PAGE, "Accéder aux Currencies")`,

- **Ligne 2209:**

```
createPermission(Resource.LEGAL, Action.ACCESS, PermissionScope.PAGE, "Accéder au Legal") ,
```

- **Ligne 2210:** `createPermission(Resource.PERMISSION, Action.ACCESS, PermissionScope.PAGE, "Accéder aux Permissions")`,

- **Ligne 2211:** `createPermission(Resource.ROLE, Action.ACCESS, PermissionScope.PAGE, "Accéder aux Roles")`,
- **Ligne 2212:**
`createPermission(Resource.TENANT, Action.ACCESS, PermissionScope.PAGE, "Accéder au Tenant")`,
- **Ligne 2213:** `createPermission(Resource.WEBHOOK, Action.ACCESS, PermissionScope.PAGE, "Accéder aux Webhooks")`,
- **Ligne 2214:** `createPermission(Resource.ANALYTIC, Action.ACCESS, PermissionScope.PAGE, "Accéder aux Analytics")`,
- **Ligne 2215:** `createPermission(Resource.IMPORATION, Action.ACCESS, PermissionScope.PAGE, "Accéder à l'Impersonation")`,
- **Ligne 2216:** `createPermission(Resource.FEATURE, Action.ACCESS, PermissionScope.PAGE, "Accéder aux Features")`,
- **Ligne 2217:** `createPermission(Resource.SUBSCRIPTION, Action.ACCESS, PermissionScope.PAGE, "Accéder aux Subscriptions")`,
- **Ligne 2218:**
`createPermission(Resource.TENANT, Action.ACCESS, PermissionScope.PAGE, "Accéder aux Tenants")`,
- **Ligne 2219:** `createPermission(Resource.TENANT_DETAIL, Action.ACCESS, PermissionScope.PAGE, "Accéder aux Détails du Tenant")`,
- **Ligne 2220:**
`createPermission(Resource.SUPERADMIN_USER, Action.ACCESS, PermissionScope.PAGE, "Accéder aux Superadmin Users")`,
- **Ligne 2231:** "Créer une demande de fonctionnalité",
- **Ligne 2232:** "Soumettre une nouvelle demande de fonctionnalité ou modification",
- **Ligne 2241:** "Voir ses propres demandes",
- **Ligne 2242:** "Consulter ses propres demandes de fonctionnalités",
- **Ligne 2252:** "Voir la liste de ses demandes de fonctionnalités",
- **Ligne 2261:** "Voir toutes les demandes",
- **Ligne 2262:** "Voir toutes les demandes de fonctionnalités du tenant",
- **Ligne 2271:** "Gérer la plateforme",
- **Ligne 2272:** "Approuver, rejeter et gérer les demandes de fonctionnalités",
- **Ligne 2281:** "Modifier les demandes",
- **Ligne 2282:** "Modifier et gérer les demandes de fonctionnalités",
- **Ligne 2291:** "Confirmer les demandes",
- **Ligne 2292:** "Confirmer et valider les demandes de fonctionnalités",
- **Ligne 2301:** "Rejeter les demandes",
- **Ligne 2302:** "Rejeter les demandes de fonctionnalités avec raison",
- **Ligne 2311:** "Supprimer les demandes",
- **Ligne 2312:** "Supprimer les demandes de fonctionnalités",
- **Ligne 2321:** "Accéder aux Feature Requests",
- **Ligne 2322:** "Accéder à la page des demandes de fonctionnalités"
- **Ligne 2332:** * Map des permissions par clé
- **Ligne 2339:** * Map des permissions par ressource
- **Ligne 2350:** * Map des permissions par catégorie
- **Ligne 2362:** // HELPER FUNCTIONS POUR VÉRIFICATION DES PERMISSIONS

- **Ligne 2366:** * Type pour le contexte utilisateur
- **Ligne 2371:** permissions: string[]; // Liste des clés de permissions
- **Ligne 2378:** * Type pour le contexte de la ressource
- **Ligne 2383:** ownerId?: string; // userId qui possède la ressource
- **Ligne 2384:** createdBy?: string; // userId qui a créé la ressource
- **Ligne 2385:** assignedTo?: string; // userId assigné à la ressource
- **Ligne 2386:** agencyId?: string; // Si la ressource appartient à une agence
- **Ligne 2387:** teamId?: string; // Si la ressource appartient à une équipe
- **Ligne 2391:** * Vérifie si un utilisateur a une permission
- **Ligne 2404:** * Vérifie si un utilisateur a une permission avec contexte (ownership)
- **Ligne 2412:** // Vérifier permission globale
- **Ligne 2417:** // Si pas de contexte de ressource, vérifier juste la permission
- **Ligne 2422:** // Vérifier permission "own"
- **Ligne 2424:** // Vérifier ownership
- **Ligne 2438:** * Vérifie si un utilisateur peut effectuer une action sur une ressource spécifique
- **Ligne 2454:** reason: Permission refusée: \${resource}.\${action} ,
- **Ligne 2462:** * Filtre les ressources auxquelles un utilisateur a accès
- **Ligne 2475:** // Si permission "own", filtrer par ownership
- **Ligne 2486:** * Obtient toutes les permissions d'un rôle
- **Ligne 2495:** * Vérifie si une permission existe
- **Ligne 2502:** * Obtient une permission par sa clé
- **Ligne 2509:** * Obtient toutes les permissions pour une ressource
- **Ligne 2516:** * Obtient toutes les clés de permissions
- **Ligne 2523:** * Obtient toutes les ressources disponibles
- **Ligne 2530:** * Obtient toutes les actions disponibles
- **Ligne 2537:** * Obtient tous les scopes disponibles

server/validators/simpleContract.ts

- **Ligne 2:** * Validators Zod pour le système simplifié de contrats MSA/SOW
- **Ligne 4:** * Ce fichier contient tous les schémas de validation pour les endpoints
- **Ligne 18:** // SCHÉMAS DE BASE
- **Ligne 22:** * Schéma pour la validation d'un fichier PDF encodé en base64
- **Ligne 26:** .min(1, "Le fichier PDF ne peut pas être vide")
- **Ligne 30:** // Vérifier que c'est du base64 valide
- **Ligne 37:** { message: "Le buffer PDF doit être encodé en base64 valide" }
- **Ligne 40:** .min(1, "Le nom du fichier est requis")
- **Ligne 41:** .max(255, "Le nom du fichier est trop long (max 255 caractères)")
- **Ligne 44:** { message: "Le fichier doit avoir l'extension .pdf" }
- **Ligne 47:** errorMap: () => ({ message: "Seuls les fichiers PDF sont acceptés" }),
- **Ligne 50:** .int("La taille du fichier doit être un entier")
- **Ligne 51:** .positive("La taille du fichier doit être positive")
- **Ligne 52:** .max(MAX_PDF_SIZE, Le fichier est trop volumineux (max \${MAX_PDF_SIZE / 1024 / 1024} MB)),

- **Ligne 56:** * Schéma pour un participant supplémentaire
- **Ligne 57:** * Au moins un de userId ou companyId doit être fourni
- **Ligne 61:** .cuid("L'ID du user doit être un CUID valide")
- **Ligne 64:** .cuid("L'ID de la company doit être un CUID valide")
- **Ligne 67:** .min(1, "Le rôle est requis")
- **Ligne 68:** .max(50, "Le rôle est trop long (max 50 caractères)")
- **Ligne 74:** message: "Au moins un de userId ou companyId doit être fourni",
- **Ligne 80:** * Tableau de participants supplémentaires pour la création de contrats
- **Ligne 87:** // SCHÉMAS POUR LES ENDPOINTS
- **Ligne 93:** * Input: PDF + informations minimales + participants supplémentaires
- **Ligne 94:** * Output: Contrat MSA créé avec statut "draft"
- **Ligne 98:** .cuid("L'ID de la company doit être un CUID valide")
- **Ligne 106:** * Input: PDF + MSA parent + informations minimales + participants supplémentaires
- **Ligne 107:** * Output: Contrat SOW créé avec statut "draft"
- **Ligne 111:** .cuid("L'ID du MSA parent doit être un CUID valide")
- **Ligne 112:** .min(1, "L'ID du MSA parent est requis"),
- **Ligne 114:** .cuid("L'ID de la company doit être un CUID valide")
- **Ligne 126:** .cuid("L'ID du contrat doit être un CUID valide")
- **Ligne 127:** .min(1, "L'ID du contrat est requis"),
- **Ligne 129:** .max(5000, "Les notes sont trop longues (max 5000 caractères)")
- **Ligne 140:** .cuid("L'ID du contrat doit être un CUID valide")
- **Ligne 141:** .min(1, "L'ID du contrat est requis"),
- **Ligne 143:** .max(5000, "Les notes sont trop longues (max 5000 caractères)")
- **Ligne 154:** .cuid("L'ID du contrat doit être un CUID valide")
- **Ligne 155:** .min(1, "L'ID du contrat est requis"),
- **Ligne 157:** .min(10, "La raison du rejet doit contenir au moins 10 caractères")
- **Ligne 158:** .max(5000, "La raison du rejet est trop longue (max 5000 caractères)"),
- **Ligne 164:** * Upload d'une version signée du contrat (completed/active)
- **Ligne 168:** .cuid("L'ID du contrat doit être un CUID valide")
- **Ligne 169:** .min(1, "L'ID du contrat est requis"),
- **Ligne 171:** .min(1, "Le fichier PDF ne peut pas être vide"),
- **Ligne 173:** .min(1, "Le nom du fichier est requis")
- **Ligne 174:** .max(255, "Le nom du fichier est trop long")
- **Ligne 177:** { message: "Le fichier doit avoir l'extension .pdf" }
- **Ligne 183:** .max(MAX_PDF_SIZE, Le fichier est trop volumineux (max \${MAX_PDF_SIZE / 1024 / 1024} MB)),
- **Ligne 193:** .cuid("L'ID du contrat doit être un CUID valide")
- **Ligne 194:** .min(1, "L'ID du contrat est requis"),
- **Ligne 196:** .max(5000, "Les notes sont trop longues (max 5000 caractères)")
- **Ligne 201:** * 7B. UPDATE SIMPLE CONTRACT (TITRE ET DESCRIPTION)
- **Ligne 203:** * Permet de mettre à jour le titre et la description d'un contrat MSA/SOW/NORM
- **Ligne 207:** .cuid("L'ID du contrat doit être un CUID valide")
- **Ligne 208:** .min(1, "L'ID du contrat est requis"),

- **Ligne 210:** `.min(1, "Le titre est requis")`
- **Ligne 211:** `.max(200, "Le titre est trop long (max 200 caractères)")`
- **Ligne 214:** `.max(1000, "La description est trop longue (max 1000 caractères)")`
- **Ligne 221:** * Filtres et pagination pour la liste des contrats
- **Ligne 242:** * Récupération d'un contrat par son ID
- **Ligne 246:** `.cuid("L'ID du contrat doit être un CUID valide")`
- **Ligne 247:** `.min(1, "L'ID du contrat est requis"),`
- **Ligne 253:** * Suppression d'un contrat en draft uniquement
- **Ligne 257:** `.cuid("L'ID du contrat doit être un CUID valide")`
- **Ligne 258:** `.min(1, "L'ID du contrat est requis"),`
- **Ligne 262:** // TYPES EXPORTÉS (inférés depuis les schémas)
- **Ligne 282:** * Schéma de base pour les champs communs des contrats NORM
- **Ligne 287:** `.cuid("L'ID de la company tenant doit être un CUID valide"),`
- **Ligne 289:** `.cuid("L'ID de l'agency doit être un CUID valide"),`
- **Ligne 291:** `.cuid("L'ID du contractor doit être un CUID valide"),`
- **Ligne 304:** `message: "Le type de salaire doit être: gross, payroll, payroll_we_pay ou split"`
- **Ligne 310:** `payrollUserId: z.string().cuid().optional(), // Pour Payroll et Payroll We Pay`
- **Ligne 315:** `.positive("Le montant du taux doit être positif")`
- **Ligne 318:** `.min(3, "La devise doit contenir au moins 3 caractères")`
- **Ligne 319:** `.max(3, "La devise doit contenir 3 caractères")`
- **Ligne 323:** `message: "Le cycle doit être: daily, weekly, monthly, yearly ou hourly"`
- **Ligne 329:** `.positive("Le montant de la marge doit être positif")`
- **Ligne 332:** `.min(3, "La devise doit contenir au moins 3 caractères")`
- **Ligne 333:** `.max(3, "La devise doit contenir 3 caractères")`
- **Ligne 336:** `errorMap: () => ({ message: "Le type de marge doit être: fixed ou percentage" }),`
- **Ligne 339:** `errorMap: () => ({ message: "La marge doit être payée par: client ou agency" }),`
- **Ligne 354:** `.int("Le nombre de jours doit être un entier")`
- **Ligne 355:** `.positive("Le nombre de jours doit être positif")`
- **Ligne 356:** `.max(365, "Le nombre de jours ne peut pas dépasser 365")`
- **Ligne 359:** `.max(5000, "Les notes sont trop longues (max 5000 caractères)")`
- **Ligne 362:** `.max(255, "La référence est trop longue (max 255 caractères)")`
- **Ligne 365:** `.min(0, "Le taux de TVA doit être entre 0 et 100")`
- **Ligne 366:** `.max(100, "Le taux de TVA doit être entre 0 et 100")`
- **Ligne 369:** `.cuid("L'ID du pays doit être un CUID valide")`
- **Ligne 382:** * Crée un contrat NORM avec validation conditionnelle selon salaryType
- **Ligne 398:** `message: "La date de début doit être antérieure à la date de fin",`
- **Ligne 417:** `message: "Champ requis selon le type de salaire sélectionné",`
- **Ligne 425:** * Met à jour un contrat NORM (draft uniquement)
- **Ligne 426:** * Tous les champs sont optionnels sauf contractId
- **Ligne 430:** `.cuid("L'ID du contrat doit être un CUID valide")`
- **Ligne 431:** `.min(1, "L'ID du contrat est requis"),`
- **Ligne 433:** // Tous les champs optionnels

- **Ligne 474:** // Validation des dates si les deux sont présentes
- **Ligne 481:** message: "La date de début doit être antérieure à la date de fin",
- **Ligne 489:** * Permet au contractor de signer son contrat
- **Ligne 493:** .cuid("L'ID du contrat doit être un CUID valide")
- **Ligne 494:** .min(1, "L'ID du contrat est requis"),
- **Ligne 498:** .optional(), // Si non fourni, on utilise la date actuelle
- **Ligne 502:** // TYPES EXPORTÉS POUR NORM
- **Ligne 516:** * Ajouter un participant à un contrat existant
- **Ligne 520:** .cuid("L'ID du contrat doit être un CUID valide")
- **Ligne 521:** .min(1, "L'ID du contrat est requis"),
- **Ligne 523:** .cuid("L'ID du user doit être un CUID valide")
- **Ligne 526:** .cuid("L'ID de la company doit être un CUID valide")
- **Ligne 529:** .min(1, "Le rôle est requis")
- **Ligne 530:** .max(50, "Le rôle est trop long (max 50 caractères)")
- **Ligne 536:** message: "Au moins un de userId ou companyId doit être fourni",
- **Ligne 544:** * Supprimer un participant d'un contrat
- **Ligne 548:** .cuid("L'ID du participant doit être un CUID valide")
- **Ligne 549:** .min(1, "L'ID du participant est requis"),
- **Ligne 555:** * Lister tous les participants d'un contrat
- **Ligne 559:** .cuid("L'ID du contrat doit être un CUID valide")
- **Ligne 560:** .min(1, "L'ID du contrat est requis"),
- **Ligne 568:** * Catégories de documents disponibles
- **Ligne 581:** * Uploader un document pour un contrat
- **Ligne 585:** .cuid("L'ID du contrat doit être un CUID valide")
- **Ligne 586:** .min(1, "L'ID du contrat est requis"),
- **Ligne 588:** .min(1, "Le fichier ne peut pas être vide"),
- **Ligne 590:** .min(1, "Le nom du fichier est requis")
- **Ligne 591:** .max(255, "Le nom du fichier est trop long (max 255 caractères)'),
- **Ligne 593:** .min(1, "Le type MIME est requis"),
- **Ligne 595:** .int("La taille du fichier doit être un entier")
- **Ligne 596:** .positive("La taille du fichier doit être positive")
- **Ligne 597:** .max(MAX_PDF_SIZE, Le fichier est trop volumineux (max \${MAX_PDF_SIZE / 1024 / 1024} MB)),
- **Ligne 599:** .min(1, "La description est requise")
- **Ligne 600:** .max(500, "La description est trop longue (max 500 caractères)'),
- **Ligne 603:** .max(1000, "Les notes sont trop longues (max 1000 caractères)")
- **Ligne 610:** * Lister tous les documents d'un contrat
- **Ligne 614:** .cuid("L'ID du contrat doit être un CUID valide")
- **Ligne 615:** .min(1, "L'ID du contrat est requis"),
- **Ligne 621:** * Supprimer un document
- **Ligne 625:** .cuid("L'ID du document doit être un CUID valide")
- **Ligne 626:** .min(1, "L'ID du document est requis"),
- **Ligne 632:** * Obtenir l'URL signée pour télécharger un document
- **Ligne 636:** .cuid("L'ID du document doit être un CUID valide")

- **Ligne 637:** .min(1, "L'ID du document est requis"),
- **Ligne 641:** // TYPES EXPORTÉS POUR PARTICIPANTS ET DOCUMENTS