Documentation Technique - Système de Gestion des Emplois du Temps

🏗️ Architecture du Système

Structure MVC Laravel

```
app/
├── Http/Controllers/
│   ├── EmploiTempsController.php     # Gestion des emplois du temps
│   ├── SubjectTeacherController.php  # Association prof-matières
│   └── Web/
│       ├── TeacherController.php     # Gestion des professeurs
│       └── ClasseController.php      # Gestion des classes
├── Models/
│   ├── EmploiTemps.php               # Modèle principal
│   ├── EmploiHoraire.php             # Liaison horaires multiples
│   ├── SubjectTeacher.php            # Association prof-matières
│   ├── Teacher.php                   # Professeurs
│   ├── Subject.php                   # Matières
│   ├── Classe.php                    # Classes
│   └── [autres modèles...]
└── database/migrations/              # Migrations de BDD
```

Relations Entre Entités

Relations Principales

```
// EmploiTemps (1) → (N) EmploiHoraire → (1) Horaire
EmploiTemps::horairess() // Horaires multiples via table pivot

// SubjectTeacher - Table pivot étendue
Teacher::subjects() // Matières du professeur
Subject::teachers() // Professeurs de la matière
Classe::subjectTeachers() // Associations de la classe

// Hiérarchie académique
Departement (1) → (N) Speciality (1) → (N) Subject
Niveauformation (1) → (N) Speciality (1) → (N) Classe
```

Contraintes d'Intégrité

```
-- Contraintes clés étrangères
ALTER TABLE emplois_temps ADD CONSTRAINT fk_emploi_classe
FOREIGN KEY (class_id) REFERENCES classes(id) ON DELETE CASCADE;

ALTER TABLE emploi_horaire ADD CONSTRAINT fk_emploi_horaire
FOREIGN KEY (emploi_temps_id) REFERENCES emplois_temps(id) ON DELETE CASCADE;

-- Index pour performances
CREATE INDEX idx_subject_teacher_lookup ON subject_teacher (class_id, trimester_id, annee_id);
CREATE INDEX idx_emploi_temps_search ON emplois_temps (class_id, trimester_id, jour_id);
```

🔧 Fonctionnalités Techniques Avancées

1. Chargement Dynamique AJAX

Chargement des Professeurs

```
// resources/views/admin/emplois/create.blade.php
function loadTeachersSubjects(row) {
    let classId = $('#class_classe_id').val();
    let trimesterId = $('#trimester_create_id').val();

    $.ajax({
        url: baseUrl + '/admin/emplois/get-teachers',
        data: { class_id: classId, trimester_id: trimesterId },
        success: function (response) {
            // Peuplement dynamique des selects
        }
    });
}
```

Endpoint API

```
// EmploiTempsController.php
public function getTeachers(Request $request) {
    $subjectTeachers = SubjectTeacher::where('trimester_id', $trimesterId)
                                     ->where('annee_id', $anneeId)
                                     ->where('class_id', $classId)
                                     ->with(['teacher', 'subject'])
                                     ->get();

    return response()->json(['data' => $formattedTeachers]);
}
```

2. Validation Avancée

Validation Multi-Niveaux

```php
// EmploiTempsController@store
// 1. Validation des champs de base
$request->validate([
    'horaire_id' => 'required|array',
    'horaire_id.*' => 'required|array',
    'horaire_id.*.*' => 'required|exists:horaires,id',
]);

// 2. Validation des conflits métier
foreach ($horaireIds as $horaireId) {
    // Vérification des conflits professeur-matière-horaire
    $conflictSameTeacherSubject = EmploiTemps::where('teacher_id', $teacherId)
        ->where('subject_id', $subjectId)
        ->where('jour_id', $jourId)
        ->whereHas('horairess', function($query) use ($horaireId) {
            $query->where('horaire_id', $horaireId);
        })
        ->exists();
}
```

3. Interface Select2 Avancée

Configuration Select2

```javascript
function initSelect2(container) {
    container.find('.horaire-select-multiple').select2({
        theme: 'bootstrap-5',
        placeholder: 'Sélectionnez les horaires...',
        allowClear: true,
        closeOnSelect: false,
        width: '100%'
    });
}
```

Gestion Dynamique des Lignes

```javascript
$('#add-row').on('click', function () {
    // Clonage et mise à jour des indices
    let rowIndex = $('.emploi-row').length;
    newRow.find('.horaire-select-multiple').attr('name', `horaire_id[${rowIndex}][]`);

    // Réinitialisation Select2
    newRow.find('.select2-container').remove();
    initSelect2(newRow);
});
```

📊 Schéma de Base de Données Détaillé

Tables Principales

emplois_temps

```sql
CREATE TABLE emplois_temps (
    id BIGINT PRIMARY KEY AUTO_INCREMENT,
    class_id BIGINT NOT NULL,
    subject_id BIGINT NOT NULL,
    teacher_id BIGINT NOT NULL,
    trimester_id BIGINT NOT NULL,
    annee_id BIGINT NOT NULL,
    jour_id BIGINT NOT NULL,
    salle_de_classe_id BIGINT NULL,
    created_at TIMESTAMP,
    updated_at TIMESTAMP,

    FOREIGN KEY (class_id) REFERENCES classes(id) ON DELETE CASCADE,
    FOREIGN KEY (subject_id) REFERENCES subjects(id) ON DELETE CASCADE,
    FOREIGN KEY (teacher_id) REFERENCES teachers(id) ON DELETE CASCADE
);
```

emploi_horaire (Table de liaison)

```sql
CREATE TABLE emploi_horaire (
    id BIGINT PRIMARY KEY AUTO_INCREMENT,
    emploi_temps_id BIGINT NOT NULL,
    horaire_id BIGINT NOT NULL,
    created_at TIMESTAMP,
    updated_at TIMESTAMP,

    FOREIGN KEY (emploi_temps_id) REFERENCES emplois_temps(id) ON DELETE CASCADE,
    FOREIGN KEY (horaire_id) REFERENCES horaires(id) ON DELETE CASCADE,
    UNIQUE KEY unique_emploi_horaire (emploi_temps_id, horaire_id)
);
```

subject_teacher (Association étendue)

```sql
CREATE TABLE subject_teacher (
    id BIGINT PRIMARY KEY AUTO_INCREMENT,
    teacher_id BIGINT NOT NULL,
    subject_id BIGINT NOT NULL,
    class_id BIGINT NOT NULL,         -- Ajout crucial
    trimester_id BIGINT NOT NULL,
    annee_id BIGINT NOT NULL,
    created_at TIMESTAMP,
    updated_at TIMESTAMP,

    UNIQUE KEY unique_assignment (teacher_id, subject_id, class_id, trimester_id, annee_id)
);
```

🔍 Requêtes Optimisées

Récupération des Emplois du Temps

```php
// Avec eager loading pour éviter le problème N+1
$emplois = EmploiTemps::with([
    'classe', 'subject', 'teacher', 'jour',
    'horairess', 'trimester', 'annee', 'salle'
])->get();

// Avec contraintes optimisées
$teachers = SubjectTeacher::where('trimester_id', $trimesterId)
                        ->where('annee_id', $anneeId)
                        ->where('class_id', $classId)
                        ->with(['teacher', 'subject'])
                        ->get();
```

Index Recommandés

```sql
-- Performance des recherches
CREATE INDEX idx_emploi_lookup ON emplois_temps (class_id, trimester_id, jour_id, annee_id);
CREATE INDEX idx_subject_teacher_filter ON subject_teacher (class_id, trimester_id, annee_id);
CREATE INDEX idx_emploi_horaire_link ON emploi_horaire (emploi_temps_id);

-- Contraintes d'unicité
CREATE UNIQUE INDEX unique_emploi_horaire ON emploi_horaire (emploi_temps_id, horaire_id);
```

Optimisations et Performances

Cache Strategy

```php
// Cache des données fréquemment utilisées
$activeYear = Cache::remember('active_year', 3600, function () {
    return Anneescolaire::where('is_active', true)->first();
});

$horaires = Cache::remember('horaires_list', 1800, function () {
    return Horaire::orderBy('ordre')->get();
});
```

Pagination Optimisée

```php
// Controller avec pagination server-side
$emplois = $query->orderBy($sort, $order)
                ->skip($offset)
                ->take($limit)
                ->get();

return response()->json([
    'total' => $total,
    'rows' => $rows,
]);
```

Sécurité et Bonnes Pratiques

Validation CSRF

```blade
<!-- Tous les formulaires incluent le token CSRF -->
<form method="POST" action="{{ route('web.emplois.store') }}">
    @csrf
    <!-- Champs du formulaire -->
</form>
```

Sanitisation des Données

```php
// Validation stricte des entrées
$request->validate([
    'class_id' => 'required|exists:classes,id',
    'teacher_id.*' => 'required|exists:teachers,id',
    'subject_id.*' => 'required|exists:subjects,id',
]);

// Échappement automatique dans Blade
{{ $teacher->name }} <!-- Sécurisé automatiquement -->
```

Gestion des Erreurs

```php
try {
    // Opérations de base de données
    $emploi = EmploiTemps::create($data);
} catch (\Exception $e) {
    Log::error('Erreur création emploi du temps', [
        'error' => $e->getMessage(),
        'data' => $data
    ]);

    return redirect()->back()
           ->withErrors(['error' => 'Erreur lors de la création'])
           ->withInput();
}
```

Responsive Design

Framework CSS

- **Bootstrap 5** pour la grille responsive
- **Select2** avec thème Bootstrap
- **Font Awesome** pour les icônes

Adaptation Mobile

```
/* Styles responsive personnalisés */
@media (max-width: 768px) {
    .emploi-row .col-md-2,
    .emploi-row .col-md-3 {
        margin-bottom: 15px;
    }

    .select2-container {
        width: 100% !important;
    }
}
```

🔧 Maintenance et Debugging

Logs Applicatifs

```
// Debugging dans EmploiTempsController
\Log::info('getTeachers called', [
    'class_id' => $request->class_id,
    'trimester_id' => $request->trimester_id,
    'result_count' => $teachers->count()
]);
```

Commandes Artisan Utiles

```
# Vider le cache des routes
php artisan route:clear

# Régénérer les classes
php artisan clear-compiled

# Migrations avec données de test
php artisan migrate:fresh --seed

# Vérification des routes
php artisan route:list --path=emplois
```

Tests de Performance

```
// Fichiers de test inclus
- test_final_improvements.php      # Tests fonctionnels
- test_select2_horaires.php       # Tests interface
- test_multiple_horaires_system.php # Tests horaires multiples
```

🔲 Évolutions Futures Possibles

Fonctionnalités Avancées

1. **Export PDF** des emplois du temps
2. **Notifications** de conflits en temps réel
3. **API REST** pour applications mobiles
4. **Dashboard** de statistiques
5. **Import/Export Excel** des données

Optimisations Techniques

1. **Queue System** pour les opérations lourdes
2. **WebSockets** pour les mises à jour temps réel
3. **Redis Cache** pour les performances
4. **Elasticsearch** pour la recherche avancée

Cette documentation technique complète le guide de configuration et fournit tous les détails nécessaires pour maintenir et développer le système.