Technical Document: Leave Rule Settings and Grant Leave Configuration Form

Overview:

This document explains the structure and functionality of the "Leave Rule Settings" and "Grant Leave" configuration form built with Angular. The form is used to set up leave-related rules, including leave codes, descriptions, and the configuration for grant frequency, auto-scheduling, and other parameters.

Technology Stack:

- **Frontend Framework**: Angular
- **UI Framework**: Bootstrap 4 (for grid layout and responsiveness)
- **Form Handling**: Angular Forms with ngModel
- **Input Types**: Checkboxes, Text Inputs, Date Pickers, and Dropdowns (ng-select2)

Form Structure and Components:

1. Leave Rule Settings Section

This section of the form manages leave-related settings for configuring leave codes, descriptions, applicability, dates, and other details.

1.1. Leave Code

- **Type**: Dropdown (ng-select2)
- **Purpose**: Select the leave code from a predefined list.
- **Behavior**: Displays a list of leave codes, and when selected, triggers a method to get the description for the leave code.

```
<label>Leave Code</label><ng-select2 [placeholder]="'Select Type'"
[data]="leaveCodeList"
   [ngModelOptions]="{standalone: true}"
(ngModelChange)="getLeaveDesc($event)"
   [disabled]="hideSubmitButton" [(ngModel)]="leaveCode"></ng-select2><small
*ngIf="leaveRuleForm.controls['leaveCode'].hasError('required') &&
(leaveRuleForm.controls['leaveCode'].touched ||
leaveRuleForm.controls['leaveCode'].dirty)" class="text-danger">
   Leave Code Required</small>
```

1.2. Leave Description

- **Type**: Read-Only Text Input
- **Purpose**: Displays the description of the selected leave code.
- **Behavior**: The field is disabled and displays the leave description when a leave code is selected.

```
<label>Leave Description</label><input type="text" class="form-control
leaveDescription" readonly
   [ngModelOptions]="{standalone: true}"
[(ngModel)]="leaveDescription" placeholder="Leave
Description"><small
*ngIf="leaveRuleForm.controls['leaveDescription'].hasError('required')
&& (leaveRuleForm.controls['leaveDescription'].touched ||
leaveRuleForm.controls['leaveDescription'].dirty)" class="text-danger">
   Leave Description Required</small>
```

1.3. Applicable For

- **Type**: Dropdown (ng-select2)
- **Purpose**: Select the gender or category applicable for this leave rule.
- **Behavior**: The field allows for selecting a gender or other applicable options.

```
<label>Applicable For</label><ng-select2 [placeholder]="'Select
Applicable'" [data]="genderList"
   [disabled]="hideSubmitButton" [ngModelOptions]="{standalone:
true}"
   [(ngModel)]="gender"></ng-select2><small
*ngIf="leaveRuleForm.controls['gender'].hasError('required') &&
(leaveRuleForm.controls['gender'].touched ||
leaveRuleForm.controls['gender'].dirty)" class="text-danger">
   Applicable Required</small>
```

1.4. From Date and To Date

- **Type**: Date Picker
- **Purpose**: Set the leave duration using the from and to dates.
- **Behavior**: The "From Date" is selected first, and the "To Date" can’t be earlier than the "From Date."

```
<label>From Date</label>&nbsp;&nbsp;&nbsp;<span
*ngIf="prevFromDate">{{ "Previous From Date " + ((prevFromDate |
date: 'dd-MM-YYYY') || "NA") }}</span><input type="date"
class="form-control" [placeholder]="'From Date'" [min]="currentDate"
[(ngModel)]="fromDate" [ngModelOptions]="{standalone: true}">
```

```
<label>To Date</label>&nbsp;&nbsp;&nbsp;<span
*ngIf="prevToDate">{{ "Previous To Date " + ((prevToDate | date: 'dd-
MM-YYYY') || "NA") }}</span><input type="date" class="form-
control" [placeholder]="'To Date'" [min]="fromDate"
[(ngModel)]="toDate">
```

1.5. Short Description

- **Type**: Text Input
- **Purpose**: Provides a brief description of the leave.

```
<label>Short Description</label><input type="text" class="form-control"
[placeholder]="'Short Description'" disabled
[ngModelOptions]="{standalone: true}" [(ngModel)]="shortDescription">
```

1.6. Number of Days

- **Type**: Text Input
- **Purpose**: Specifies the number of days for the leave.
- **Behavior**: Only numeric input is allowed.

```
<label>Number of Days</label><input type="text" class="form-control"
[placeholder]="'Number of Days'" onkeypress='return
(event.charCode >= 48 && event.charCode <= 57) || event.charCode <=
31' [ngModelOptions]="{standalone: true}"
[(ngModel)]="numberOfDays">
```

1.7. Minimum Apply Count

- **Type**: Text Input
- **Purpose**: Specifies the minimum number of days required to apply for leave.
- **Behavior**: Only numeric input is allowed.

```
<label>Minimum Apply Count</label><input type="text" class="form-
control" [placeholder]="'Minimum Apply Count'" onkeypress='return
```

```
(event.charCode >= 48 && event.charCode <= 57) || event.charCode <=
31' [ngModelOptions]="{standalone: true}"
[(ngModel)]="minApplyCount">
```

2. Grant Leave Section

This section configures how leave is granted, including the frequency, count, and whether the leave is auto-scheduled.

2.1. Grant Frequency

- **Type**: Dropdown (ng-select)
- **Purpose**: Select the frequency at which the leave is granted.

```
<label>Grant Frequency</label><ng-select [placeholder]="'Grant
Frequency'" [items]="grantFrequencyList"
[(ngModel)]="grantFrequency" style="z-index: 10000;" bindLabel="text"
bindValue="id" [ngModelOptions]="{standalone: true}"></ng-select>
```

2.2. Grant Leave Count

- **Type**: Text Input
- **Purpose**: Specify the count of leave to be granted.
- **Behavior**: Only numeric input is allowed.

```
<label>Grant Leave Count</label><small class="text-mandatory">
*</small><input type="text" class="form-control" [placeholder]="'Grant
Leave Count'" onkeypress='return (event.charCode >= 48 &&
event.charCode <= 57) || event.charCode <= 31'
[ngModelOptions]="{standalone: true}" [(ngModel)]="grantLeaveCount">
```

2.3. Grant Auto Schedule

- **Type**: Checkbox
- **Purpose**: Determine if the leave should be auto-scheduled.
- **Behavior**: When enabled, the leave is automatically scheduled.

```
<label>Grant Auto Schedule</label><small class="text-mandatory">
*</small><br><label class="switch switch-flat">
```

```
    <input class="switch-input" type="checkbox"
[checked]="grantAutoSchedule" [(ngModel)]="grantAutoSchedule"
(change)="grantAutoScheduleChange($event)" />
    <span class="switch-label" data-on="Yes" data-off="No"></span>
    <span class="switch-handle"></span></label>
```

Key Angular Features and Binding:

- **[(ngModel)]**: Two-way data binding between the input fields and the component properties.
- **(ngModelChange)**: Detects changes to the ng-select2 component and triggers related methods.
- **[ngModelOptions]**: Provides additional configuration options, such as standalone binding.
- **(change)**: Event binding to detect changes in checkbox inputs, triggering relevant methods in the component.
- **ng-select/ng-select2**: Custom dropdown components with enhanced features like searchability and custom styling.

Conclusion:

This form allows administrators to configure leave rules and grant leave settings. Angular's features, such as ngModel for two-way binding, ng-select for enhanced dropdown functionality, and event binding for dynamic form interaction, provide a robust and flexible system for leave management.

Technical Document: Leave Policy Settings Form

Overview: This document provides a technical overview of the Leave Policy Settings form, which allows users to configure various leave-related rules in the system. The form handles the configuration of restrictions, conditions, and limits related to employee leave policies, such as post-resignation leave restrictions, committee approval, and limits on leave usage.

Technology Stack:

- **Frontend Framework:** Angular
- **UI Framework:** Bootstrap 4 (for grid layout and responsiveness)
- **Form Handling:** Angular Forms with ngModel
- **Input Types:** Checkboxes, Text Inputs, Dropdowns (ng-select), Switches

---

Form Structure and Components:

1. After Resignation Restriction

- **Type:** Checkbox & Text Input
- **Purpose:** Allows setting a restriction on leave after resignation. The restriction can be activated by the checkbox, and a numeric input field becomes visible when the restriction is enabled.
- **Behavior:**
    - When the checkbox is toggled on, the input field for setting the restriction limit becomes visible.
    - Input is restricted to numeric values.

HTML Example:

```
<label>After Resignation Restriction <small class="text-mandatory">*</small></label><label class="switch switch-flat">
    <input class="switch-input" type="checkbox"
        [checked]="afterResignationRestrictionActive"
        [(ngModel)]="afterResignationRestrictionActive"
        (change)="afterResignationRestrictionChange($event)" />
    <span class="switch-label" data-on="Yes" data-off="No"></span>
    <span class="switch-handle"></span></label><div
*ngIf="afterResignationRestrictionActive">
```

```
    <input type="text" class="form-control" placeholder="After Resignation Restriction"
        onkeypress='return (event.charCode >= 48 && event.charCode <= 57) || event.charCode <= 31'
        [(ngModel)]="afterResignationRestriction" required /></div>
```

---

2. Apply Only on Specific Dates

- **Type:** Checkbox & Dropdown (ng-select)
- **Purpose:** Enables the option to apply leave restrictions only on specific dates. When the checkbox is checked, a dropdown to select leave types is shown.
- **Behavior:**
    - When activated, the user can choose from a list of available leave types (e.g., vacation, sick leave).

HTML Example:

```
<label>Apply Only on Specific Dates <small class="text-mandatory">*</small></label><label class="switch switch-flat">
    <input class="switch-input" type="checkbox"
        [checked]="isApplyOnSpecificDatesActive"
        [(ngModel)]="isApplyOnSpecificDatesActive"
        (change)="isApplyOnSpecificDatesActiveChange($event)" />
    <span class="switch-label" data-on="Yes" data-off="No"></span>
    <span class="switch-handle"></span></label><div *ngIf="isApplyOnSpecificDatesActive">
    <ng-select [placeholder]="'Select'" [items]="specificLeaveCodeList"
        [(ngModel)]="applyOnSpecificDates" style="z-index: 10000;"
        bindLabel="text" bindValue="id" [ngModelOptions]="{standalone: true}">
    </ng-select></div>
```

---

3. Prior Leave Apply

- **Type:** Checkbox & Text Input
- **Purpose:** Configures whether prior leave must be applied for. When enabled, a field appears to set the count of prior leave days.
- **Behavior:**

    - When enabled, the input field to specify the number of prior leave days becomes visible.

HTML Example:

```
<label>Prior Leave Apply <small class="text-
mandatory">*</small></label><label class="switch switch-flat">
  <input class="switch-input" type="checkbox"
    [checked]="isPriorActive"
    [(ngModel)]="isPriorActive"
    (change)="isPriorActiveChange($event)" />
  <span class="switch-label" data-on="Yes" data-off="No"></span>
  <span class="switch-handle"></span></label><div
*ngIf="isPriorActive">
  <input type="text" class="form-control" placeholder="Prior Leave
Apply"
    [(ngModel)]="priorLeaveApplyCount" required /></div>
```

---

4. Committee Approval

- **Type:** Checkbox & Text Input
- **Purpose:** Configures whether committee approval is required for leave. When enabled, an input field for specifying the committee approval limit appears.
- **Behavior:**
    - When activated, the input field to set the approval details becomes visible.

HTML Example:

```
<label>Committee Approval <small class="text-
mandatory">*</small></label><label class="switch switch-flat">
  <input class="switch-input" type="checkbox"
    [checked]="isCommitteeApprovalActive"
    [(ngModel)]="isCommitteeApprovalActive"
    (change)="isCommitteeApprovalActiveChange($event)" />
  <span class="switch-label" data-on="Yes" data-off="No"></span>
  <span class="switch-handle"></span></label><div
*ngIf="isCommitteeApprovalActive">
  <input type="text" class="form-control" placeholder="Committee
Approval"
    [(ngModel)]="committeeApproval" /></div>
```

---

5. Restricted with Other Leave Types

- **Type:** Checkbox & Dropdown (ng-select)
- **Purpose:** Allows the restriction of specific leave types when other leave types are already used. Multiple leave types can be selected.
- **Behavior:**
  - The dropdown appears when the checkbox is checked, allowing multiple leave types to be selected.

HTML Example:

```
<label>Restricted with Other Leave Types <small class="text-mandatory">*</small></label><label class="switch switch-flat">
  <input class="switch-input" type="checkbox"
    [checked]="isRestrictedLeaveActive"
    [(ngModel)]="isRestrictedLeaveActive"
    (change)="isRestrictedLeaveActiveChange($event)" />
  <span class="switch-label" data-on="Yes" data-off="No"></span>
  <span class="switch-handle"></span></label><div *ngIf="isRestrictedLeaveActive">
  <ng-select [placeholder]="'Select'" [items]="leaveCodeList"
    [(ngModel)]="restrictedWithOtherLeaveTypes" [multiple]="true"
    style="z-index: 10000;" bindLabel="text" bindValue="id"
    [ngModelOptions]="{standalone: true}">
  </ng-select></div>
```

Additional Components:

- **Validity Duration:** A checkbox input that activates an input field to set the validity duration of a leave policy.
- **Limit Apply in Period:** A checkbox with two text inputs to specify the max limit and period for leave application.
- **Maximum Accumulated Leave:** A checkbox to enable the setting of a maximum accumulated leave limit, with an input field to specify the value.
- **Carry Over to Next Year:** A checkbox to specify whether leave can be carried over to the next year.
- **EN Cash:** A checkbox to enable EN cash options, with additional fields to specify the EN Cash rule and maximum leave count.

Conclusion: This form provides a flexible and dynamic interface for configuring various leave-related rules within an organization. By utilizing Angular forms with ngModel binding and conditionally displayed inputs based on checkbox states, it ensures a smooth user experience while maintaining consistency across the application.

Technical Document: Leave Rule Settings and Weekends/Holidays Clubbing Form

Overview

This document explains the structure and functionality of the "Leave Rule Settings" and "Weekends, Holidays, Clubbing, and Covering" form built with Angular. The form includes various toggles, input fields, and dropdowns designed to configure and manage leave-related settings, including leave rules, paid leave, half-day leave, leave balance checks, and how weekends and holidays interact with leave periods.

Technology Stack:

- **Frontend Framework**: Angular
- **UI Framework**: Bootstrap 4 (for grid layout and responsiveness)
- **Form Handling**: Angular Forms with ngModel
- **Input Type**: Checkboxes, Text Inputs, and Select Dropdowns (ng-select)

Form Structure and Components

1. Leave Rule Settings Section

The "Leave Rule Settings" section contains a series of form fields that allow administrators to configure various rules and requirements for leave applications. Each of these fields uses a toggle switch to enable/disable specific rules.

1.1. Proof Requirement

- **Type**: Checkbox Toggle
- **Purpose**: Determines whether proof of leave is required after resignation or any other conditions.
- **Behavior**:
    - When active, a text input field appears below it where the user can input the minimum number of leave days required for proof.
    - If deactivated, the input field is hidden.

```
<label>Proof Requirement</label><input type="checkbox"
[(ngModel)]="proofRequiredActive"
(change)="proofRequiredChange($event)" /><input
*ngIf="proofRequiredActive" type="text" placeholder="After
Resignation Restriction" />
```

1.2. Paid Leave Applicability

- **Type**: Checkbox Toggle
- **Purpose**: Enables or disables the applicability of paid leave.
- **Behavior**: When activated, it indicates that paid leave can be requested; otherwise, it is deactivated.

```
<label>Paid Leave Applicability</label><input type="checkbox"
[(ngModel)]="isPaidLeaveActive"
(change)="isPaidLeaveChange($event)" />
```

1.3. Allow Half Day Leave

- **Type**: Checkbox Toggle
- **Purpose**: Allows the system to process half-day leave requests.
- **Behavior**: When enabled, half-day leave requests are allowed; otherwise, only full-day leaves are permitted.

```
<label>Allow Half Day Leave</label><input type="checkbox"
[(ngModel)]="allowHalfDayLeaveActive"
(change)="allowHalfDayLeaveActiveChange($event)" />
```

1.4. Check Leave Balance

- **Type**: Checkbox Toggle
- **Purpose**: Ensures that users must check their leave balance before requesting leave.
- **Behavior**: When active, the leave balance is checked before proceeding with the leave request.

```
<label>Check Leave Balance</label><input type="checkbox"
[(ngModel)]="checkLeaveBalanceActive"
(change)="checkLeaveBalanceActiveChange($event)" />
```

1.5. Calculated as Working Day

- **Type**: Checkbox Toggle
- **Purpose**: Determines whether leave is calculated as a working day or not.
- **Behavior**: When active, the leave is treated as part of the working days.

```
<label>Calculated as Working Day</label><input type="checkbox"
[(ngModel)]="isCalculatedActive"
(change)="isCalculatedActiveChange($event)" />
```

---

2. Weekends, Holidays, Clubbing, and Covering Section

This section allows users to configure how weekends and holidays are handled when a leave period falls over them. It includes options for including these days as part of the leave and for clubbing them with leave requests.

2.1. Consider Weekends Between Leave as Leave

- **Type**: Checkbox Toggle
- **Purpose**: Determines if weekends between the leave period are to be considered as leave.
- **Behavior**: If activated, weekends between the leave start and end dates will be counted as part of the leave.

```
<label>Consider weekends between leave period as leave</label><input
type="checkbox" [(ngModel)]="includeWeekendsInBetweenLeave"
(change)="includeWeekendsInBetweenLeaveChange($event)" />
```

2.2. Allow Clubbing Weekends with Leave

- **Type**: Checkbox Toggle
- **Purpose**: Allows weekends to be clubbed with the leave period, thus extending the leave to include weekends.
- **Behavior**: When activated, weekends are included with the leave request.

```
<label>Allow clubbing weekends with leave</label><input
type="checkbox" [(ngModel)]="considerClubbingWeekends"
(change)="allowClubbingWeekendsChange($event)" />
```

2.3. Consider as Leave if Weekend is Clubbed

- **Type**: Dropdown (ng-select)
- **Purpose**: Allows the selection of a leave type when weekends are clubbed with leave.
- **Behavior**: The dropdown is enabled if the option to club weekends with leave is selected, and users can select from predefined leave types.

```
<ng-select [placeholder]="'Select'" [items]="leaveClubbingList"
[disabled]="!considerClubbingWeekends"
[(ngModel)]="clubWeekendsWithLeave"></ng-select>
```

2.4. Consider Holidays Between Leave as Leave

- **Type**: Checkbox Toggle
- **Purpose**: Determines if holidays between the leave period should be considered part of the leave.
- **Behavior**: When active, holidays within the leave period are counted as leave days.

```
<label>Consider Holidays between leave period as leave</label><input
type="checkbox" [(ngModel)]="includeHolidaysInBetweenLeave"
(change)="includeHolidaysInBetweenLeaveChange($event)" />
```

2.5. Allow Clubbing Holidays with Leave

- **Type**: Checkbox Toggle
- **Purpose**: Allows holidays to be clubbed with the leave.
- **Behavior**: When enabled, holidays can be combined with leave.

```
<label>Allow clubbing holidays with leave</label><input
type="checkbox" [(ngModel)]="considerClubbingHolidays"
(change)="allowClubbingHolidaysChange($event)" />
```

2.6. Consider as Leave if Holidays are Clubbed

- **Type**: Dropdown (ng-select)
- **Purpose**: Enables users to select the type of leave when holidays are clubbed with leave.
- **Behavior**: This dropdown is enabled if clubbing holidays with leave is selected.

```
<ng-select [placeholder]="'Select'" [items]="leaveClubbingList"
[disabled]="!considerClubbingHolidays"
[(ngModel)]="clubHolidaysWithLeave"></ng-select>
```

Key Angular Features and Binding

[(ngModel)]:

This is used for two-way data binding, linking the form inputs to the component properties. Any changes made to the input will update the bound property, and vice versa.

(change):

Event binding to detect changes in the checkbox or input field. This triggers methods in the component to handle the changes and update the relevant logic.

***ngIf:**

Used to conditionally render elements, such as the input for the "Proof Requirement," which only appears when the proofRequiredActive checkbox is checked.

ng-select:

This is a custom dropdown component from ng-select, allowing for enhanced select input functionality with options for searching, grouping, and styling. It is used here for selecting leave types when weekends or holidays are clubbed with leave.

Conclusion

This form provides comprehensive functionality for managing leave-related settings and clubbing rules for weekends and holidays. It uses Angular's reactive form features, such as ngModel, two-way data binding, and event handling, to dynamically change the UI based on user input. This allows for flexible and intuitive configuration of leave rules in an enterprise environment.