Date-Picker

https://playdocs1.orangeriver-ad055946.westus2.azurecontainerapps.io/play-docs/docs/ui-components/Inputs/Date-Picker

Date Picker

Thecomponent provides a robust, accessible, and highly customizable calendar interface for selecting single dates or date ranges. It features structured input fields, keyboard navigation, glassmorphism effects, and seamless integration with Angular forms. The component is designed for both standalone display and as a building block for date pickers and scheduling interfaces.

```
<aava-calendar>
```

How to use

```
import { AavaCalendarComponent } from "@aava/play-core";
```

```
import { AavaCalendarComponent } from "@aava/play-core";
```

Basic Usage

Simple calendar for single date selection with structured input fields.

```
<aava-datepicker (dateSelected)="onDateSelected($event)"> </aava-datepicker>
```

```
<aava-datepicker (dateSelected)="onDateSelected($event)"> </aava-datepicker>
```

```
onDateSelected(date: Date) {  console.log('Selected date:', date);}
```

```
onDateSelected(date: Date) {  console.log('Selected date:', date);}
```

Range Selection

Enable range mode for selecting a start and end date with dual structured inputs.

```
<aava-datepicker [isRange]="true" (rangeSelected)="onRangeSelected($event)"></aava-datepicker>
```

```
<aava-datepicker [isRange]="true" (rangeSelected)="onRangeSelected($event)"></aava-datepicker>
```

```
  selectedRange: DateRange = { start: null, end: null };  onRangeSelected(range: DateRange) {
this.selectedRange = range;    console.log('Selected range:', range);  }
```

```
  selectedRange: DateRange = { start: null, end: null };  onRangeSelected(range: DateRange) {
this.selectedRange = range;    console.log('Selected range:', range);  }
```

Always Open / Embedded Mode

Display the calendar inline (always open) for dashboards and embedded views.

```
<aava-datepicker [alwaysOpen]="true" (dateSelected)="onDateSelected($event)"></aava-datepicker>
```

```
<aava-datepicker [alwaysOpen]="true" (dateSelected)="onDateSelected($event)"></aava-datepicker>
```

```
  selectedDate: Date | null = null;  onDateSelected(date: Date) {    this.selectedDate = date;
console.log('Selected date:', date);  }
```

```
  selectedDate: Date | null = null;  onDateSelected(date: Date) {    this.selectedDate = date;
console.log('Selected date:', date);  }
```

Customization Options

Showcase selector shape and various customization options.

```
<aava-datepicker  [alwaysOpen]="true"  selectorShape="square"  (dateSelected)="onDateSelected($event)"></aava-
datepicker><aava-datepicker  [alwaysOpen]="true"  selectorShape="circle"  (dateSelected)="onDateSelected($event)">
</aava-datepicker>
```

```
<aava-datepicker  [alwaysOpen]="true"  selectorShape="square"  (dateSelected)="onDateSelected($event)"></aava-
datepicker><aava-datepicker  [alwaysOpen]="true"  selectorShape="circle"  (dateSelected)="onDateSelected($event)">
</aava-datepicker>
```

```
  onDateSelected(date: Date) {    console.log('Selected date:', date);  }
```

```
  onDateSelected(date: Date) {    console.log('Selected date:', date);  }
```

Size Variants

Five size variants to accommodate different design requirements and space constraints.

```
<aava-datepicker  (dateSelected)="onDateSelected($event)"  calendarSize="xs"  label="Extra Small"></aava-
datepicker><aava-datepicker  (dateSelected)="onDateSelected($event)"  calendarSize="sm"  label="Small"></aava-
datepicker><aava-datepicker  (dateSelected)="onDateSelected($event)"  calendarSize="md"  label="Medium"></aava-
datepicker><aava-datepicker  (dateSelected)="onDateSelected($event)"  calendarSize="lg"  label="Large"></aava-
datepicker><aava-datepicker  (dateSelected)="onDateSelected($event)"  calendarSize="xl"  label="Extra Large">
</aava-datepicker>
```

```
<aava-datepicker  (dateSelected)="onDateSelected($event)"  calendarSize="xs"  label="Extra Small"></aava-
datepicker><aava-datepicker  (dateSelected)="onDateSelected($event)"  calendarSize="sm"  label="Small"></aava-
datepicker><aava-datepicker  (dateSelected)="onDateSelected($event)"  calendarSize="md"  label="Medium"></aava-
datepicker><aava-datepicker  (dateSelected)="onDateSelected($event)"  calendarSize="lg"  label="Large"></aava-
datepicker><aava-datepicker  (dateSelected)="onDateSelected($event)"  calendarSize="xl"  label="Extra Large">
</aava-datepicker>
```

```
  onDateSelected(date: Date) {    console.log('Selected date:', date);  }
```

```
  onDateSelected(date: Date) {    console.log('Selected date:', date);  }
```

Available Sizes

• xs (Extra Small)- Compact size for tight spaces and dense layouts

• sm (Small)- Smaller than default, suitable for secondary interfaces

• md (Medium)- Default size, balanced for most use cases

• lg (Large)- Larger size for enhanced readability and touch interfaces

• xl (Extra Large)- Maximum size for accessibility and prominent displays

Size Characteristics

Each size variant affects:

• Input field dimensions- Height, padding, and font sizes

• Calendar popup sizing- Overall dimensions and spacing

• Day selector sizes- Individual day cell dimensions

• Navigation elements- Button sizes and spacing

• Typography scaling- Font sizes for headers and content

Accessibility Features

Built-in accessibility features ensuring WCAG compliance and inclusive user experience.

```
<aava-datepicker [alwaysOpen]="true" (dateSelected)="onDateSelected($event)"></aava-datepicker>
```

```
<aava-datepicker [alwaysOpen]="true" (dateSelected)="onDateSelected($event)"></aava-datepicker>
```

```
  onDateSelected(date: Date) {    console.log('Selected date:', date);  }
```

```
  onDateSelected(date: Date) {    console.log('Selected date:', date);  }
```

Features

- Single date and range selectionwith visual range indicators
- Structured input fieldsfor direct date entry (DD/MM/YYYY format)
- Multiple size variants(xs, sm, md, lg, xl) for flexible layouts
- Keyboard navigation(arrow keys, Enter, Tab, Escape)
- Month/year navigationwith dropdown selectors
- Customizable selector shape(square or circle)
- Glassmorphism surface effectswith multiple strength levels
- Weekday format options(1, 2, or 3 letter formats)
- Full accessibility(ARIA, focus management, screen reader support)
- Angular forms integration(ControlValueAccessor implementation)
- Auto-advance input segmentsfor seamless data entry
- Input validationand error handling
- Responsive designwith mobile-friendly interactions

API Reference

Inputs

```
Property | Type | Default | Description
isRange | boolean | false | Enable range selection mode
selectedDate | Date | null | null | Selected date (single mode)
dateRange | { start: Date | null, end: Date | null } | { start: null, end: null } | Selected date range (range
mode)
weekdayFormat | 1 | 2 | 3 | 3 | Weekday label format: 1 = single letter, 2 = two-letter, 3 = three-letter
alwaysOpen | boolean | false | Display calendar inline (always open)
selectorShape | 'square' | 'circle' | 'square' | Shape of day selector
calendarSize | 'xs' | 'sm' | 'md' | 'lg' | 'xl' | 'md' | Size variant for calendar and input elements
surface | boolean | false | Enable glassmorphism surface effect
surfaceStrength | 'medium' | 'strong' | 'max' | undefined | 'medium' | Glassmorphism intensity
```

```
isRange
```

```
boolean
```

```
false
```

```
selectedDate
```

```
Date | null
```

```
null
```

```
dateRange
```

```
{ start: Date | null, end: Date | null }
```

```
{ start: null, end: null }
```

```
weekdayFormat
```

```
1 | 2 | 3
```

```
3
```

```
alwaysOpen
```

```
boolean
```

```
false
```

```
selectorShape
```

```
'square' | 'circle'
```

```
'square'
```

```
calendarSize
```

```
'xs' | 'sm' | 'md' | 'lg' | 'xl'
```

```
'md'
```

```
surface
```

```
boolean
```

```
false
```

```
surfaceStrength
```

```
'medium' | 'strong' | 'max' | undefined
```

```
'medium'
```

Outputs

```
Event | Type | Description
dateSelected | EventEmitter | Emitted when a date is selected (single mode)
rangeSelected | EventEmitter<{ start: Date, end: Date }> | Emitted when a date range is selected (range mode)
```

```
dateSelected
```

```
EventEmitter<Date>
```

```
rangeSelected
```

```
EventEmitter<{ start: Date, end: Date }>
```

Computed Properties

```
Property | Type | Description
weekDays | string[] | Computed weekday labels based on format
yearRange | number[] | Array of years (current ± 50 years)
calendarDays | CalendarDay[] | Array of days for current month display
```

```
weekDays
```

```
string[]
```

```
yearRange
```

```
number[]
```

```
calendarDays
```

```
CalendarDay[]
```

Interfaces

```
interface DateRange {  start: Date | null;  end: Date | null;}interface CalendarDay {  date: Date;
isCurrentMonth: boolean;  isToday: boolean;  isSelected: boolean;  isInRange: boolean;  isRangeStart: boolean;
isRangeEnd: boolean;}
```

```
interface DateRange {  start: Date | null;  end: Date | null;}interface CalendarDay {  date: Date;
isCurrentMonth: boolean;  isToday: boolean;  isSelected: boolean;  isInRange: boolean;  isRangeStart: boolean;
isRangeEnd: boolean;}
```

CSS Custom Properties

```
Property | Description
--calendar-font-family | Font family for calendar text
--calendar-size-sm-font | Small font size for calendar elements
--calendar-size-md-font | Medium font size for headers (default)
--calendar-input-padding | Padding for input fields
--calendar-input-border | Border for input fields
--calendar-input-border-radius | Border radius for input fields
--calendar-input-background | Background color for input fields
--calendar-popup-background | Background for calendar popup
--calendar-popup-border | Border for calendar popup
--calendar-popup-border-radius | Border radius for calendar popup
--calendar-popup-shadow | Shadow for calendar popup
--calendar-popup-z-index | Z-index for calendar popup
--calendar-nav-button-hover-background | Hover background for navigation buttons
```

```
--calendar-font-family
```

```
--calendar-size-sm-font
```

```
--calendar-size-md-font
```

```
--calendar-input-padding
```

```
--calendar-input-border
```

```
--calendar-input-border-radius
```

```
--calendar-input-background
```

```
--calendar-popup-background
```

```
--calendar-popup-border
```

```
--calendar-popup-border-radius
```

```
--calendar-popup-shadow
```

```
--calendar-popup-z-index
```

```
--calendar-nav-button-hover-background
```

Accessibility

The Calendar component is built with comprehensive accessibility features:

• Semantic HTML: Uses appropriate roles and elements for calendar, grid, and buttons

• ARIA attributes: Provides ARIA labels for navigation, days, and range selection

• Keyboard navigation: Full support for arrow keys, Enter, Tab, and Escape

• Screen reader support: Announces current month/year, selected dates, and range

• Focus management: Clear focus indicators and logical tab order

• High contrast: Supports high contrast and custom themes

• Reduced motion: Respects user motion preferences

Keyboard Shortcuts

• Arrow keys: Move focus between days

• Enter/Space: Select date or range endpoint

• Tab/Shift+Tab: Move between input segments and controls

• Escape: Close popup (if not always open)

• Home/End: Navigate to first/last day of month

• Page Up/Down: Navigate to previous/next month

Structured Input Navigation

• Tab: Move between day, month, year segments

• Arrow keys: Navigate within segments

• Auto-advance: Automatically moves to next segment when complete

• Validation: Real-time validation of date inputs

Best Practices

Usage Guidelines

• Single Date: Use for appointment booking, event scheduling, or simple date selection

• Range Selection: Use for booking periods, analytics date ranges, or vacation planning

- Always Open: Use for dashboards, embedded views, or when space allows
- Size Selection: Choose appropriate size based on context and space constraints
- Surface Effects: Use glassmorphism for modern, layered UI designs
- Forms Integration: Leverage ControlValueAccessor for seamless form integration

Size Selection Guidelines

- XS: Use in data tables, compact forms, or when space is extremely limited
- SM: Use in secondary interfaces, sidebars, or when you need subtle presence
- MD: Use as the default for most applications and primary interfaces
- LG: Use in touch interfaces, mobile applications, or when enhanced readability is needed
- XL: Use in accessibility-focused applications or when maximum visibility is required

Implementation Considerations

- Input Validation: Always validate user input from structured fields
- Error Handling: Provide clear feedback for invalid dates
- Mobile Experience: Test touch interactions and responsive behavior
- Performance: Calendar renders efficiently with OnPush change detection
- Theming: Use semantic tokens for consistent design system integration

Accessibility Implementation

- Screen Reader Testing: Test with NVDA, JAWS, or VoiceOver
- Keyboard Testing: Verify complete keyboard navigation flow
- Focus Management: Ensure logical tab order and clear focus indicators
- Color Contrast: Maintain sufficient contrast ratios (4.5:1 minimum)
- Motion Preferences: Respect user's reduced motion settings