Popover

Popover

Thecomponent provides a flexible overlay system for displaying contextual information, help text, and guided tour content. Built with advanced positioning logic and multiple navigation variants, it's perfect for creating interactive user experiences.

```
<ava-popover>
```

How to use

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

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

Basic Usage

Simple popover implementation with default settings and basic content display.

```
<aava-button  avaPopover  [avaPopoverData]="samplePopoverData"  [avaPopoverPosition]="'top'"
[avaPopoverArrow]="'center'"  [label]="'Show Popover'"  [variant]="'primary'"  [outlined]="true"
pressedEffect="ripple"></aava-button><aava-button  avaPopover  [avaPopoverData]="learnMorePopoverData"
[avaPopoverPosition]="'bottom'"  [avaPopoverArrow]="'center'"  [avaPopoverShowLearnMore]="true"  [label]="'Show
with Learn More'"  [variant]="'secondary'"  [outlined]="true"  pressedEffect="ripple"></aava-button><aava-button
avaPopover  [avaPopoverData]="multiStepPopoverData"  [avaPopoverPosition]="'right'"  [avaPopoverArrow]="'center'"
[avaPopoverShowButtons]="true"  [avaPopoverShowPagination]="true"  [label]="'Show Multi-step'"
[variant]="'success'"  [outlined]="true"  pressedEffect="ripple"></aava-button>
```

```
<aava-button  avaPopover  [avaPopoverData]="samplePopoverData"  [avaPopoverPosition]="'top'"
[avaPopoverArrow]="'center'"  [label]="'Show Popover'"  [variant]="'primary'"  [outlined]="true"
pressedEffect="ripple"></aava-button><aava-button  avaPopover  [avaPopoverData]="learnMorePopoverData"
[avaPopoverPosition]="'bottom'"  [avaPopoverArrow]="'center'"  [avaPopoverShowLearnMore]="true"  [label]="'Show
with Learn More'"  [variant]="'secondary'"  [outlined]="true"  pressedEffect="ripple"></aava-button><aava-button
avaPopover  [avaPopoverData]="multiStepPopoverData"  [avaPopoverPosition]="'right'"  [avaPopoverArrow]="'center'"
[avaPopoverShowButtons]="true"  [avaPopoverShowPagination]="true"  [label]="'Show Multi-step'"
[variant]="'success'"  [outlined]="true"  pressedEffect="ripple"></aava-button>
```

```
  samplePopoverData: PopOverData[] = [    {      header: 'Welcome!',      description: 'This is a simple popover
with basic content.',    },  ]; learnMorePopoverData: PopOverData[] = [    {      header: 'Learn More',
description:        'This popover includes a learn more link for additional information.',      learnMoreUrl:
'https://example.com',    },  ]; multiStepPopoverData: PopOverData[] = [    {      header: 'Step 1',
description: 'This is the first step of the multi-step popover.',    },    {      header: 'Step 2',
description: 'This is the second step with additional information.',    },    {      header: 'Step 3',
description: 'This is the final step with completion details.',    },  ];
```

```
  samplePopoverData: PopOverData[] = [    {      header: 'Welcome!',      description: 'This is a simple popover
with basic content.',    },  ]; learnMorePopoverData: PopOverData[] = [    {      header: 'Learn More',
description:        'This popover includes a learn more link for additional information.',      learnMoreUrl:
'https://example.com',    },  ]; multiStepPopoverData: PopOverData[] = [    {      header: 'Step 1',
description: 'This is the first step of the multi-step popover.',    },    {      header: 'Step 2',
description: 'This is the second step with additional information.',    },    {      header: 'Step 3',
description: 'This is the final step with completion details.',    },  ];
```

Basic Features

• Flexible Positioning: Automatic positioning with arrow alignment

• Content Management: Support for header, description, and optional learn more links

• Responsive Design: Adapts to different screen sizes and content lengths

• Accessibility: Built-in accessibility features and keyboard navigation

- Animation: Smooth entrance animations with stretch effects

Positioning Options

Four positioning variants with intelligent arrow placement and automatic boundary detection.

```
<aava-button  avaPopover  [avaPopoverData]="topPopoverData"  [avaPopoverPosition]="'top'"
[avaPopoverArrow]="'center'"  [label]="'Show Top'"  [variant]="'primary'"  [outlined]="true"
pressedEffect="ripple"></aava-button><aava-button  avaPopover  [avaPopoverData]="bottomPopoverData"
[avaPopoverPosition]="'bottom'"  [avaPopoverArrow]="'center'"  [label]="'Show Bottom'"  [variant]="'secondary'"
[outlined]="true"  pressedEffect="ripple"></aava-button><aava-button  avaPopover
[avaPopoverData]="leftPopoverData"  [avaPopoverPosition]="'left'"  [avaPopoverArrow]="'center'"  [label]="'Show
Left'"  [variant]="'success'"  [outlined]="true"  pressedEffect="ripple"></aava-button><aava-button  avaPopover
[avaPopoverData]="rightPopoverData"  [avaPopoverPosition]="'right'"  [avaPopoverArrow]="'center'"  [label]="'Show
Right'"  [variant]="'warning'"  [outlined]="true"  pressedEffect="ripple"></aava-button><aava-button  avaPopover
[avaPopoverData]="startArrowPopoverData"  [avaPopoverPosition]="'top'"  [avaPopoverArrow]="'start'"
[label]="'Start Arrow'"  [variant]="'info'"  [outlined]="true"  pressedEffect="ripple"></aava-button><aava-button
avaPopover  [avaPopoverData]="centerArrowPopoverData"  [avaPopoverPosition]="'top'"  [avaPopoverArrow]="'center'"
[label]="'Center Arrow'"  [variant]="'primary'"  [outlined]="true"  pressedEffect="ripple"></aava-button><aava-
button  avaPopover  [avaPopoverData]="endArrowPopoverData"  [avaPopoverPosition]="'top'"
[avaPopoverArrow]="'end'"  [label]="'End Arrow'"  [variant]="'secondary'"  [outlined]="true"
pressedEffect="ripple"></aava-button>
```

```
 topPopoverData: PopOverData[] = [    {       header: 'Top Position',      description:        'This popover
appears above the target element with a downward arrow.',    },  ]; bottomPopoverData: PopOverData[] = [    {
header: 'Bottom Position',      description:        'This popover appears below the target element with an upward
arrow.',    },  ]; leftPopoverData: PopOverData[] = [    {      header: 'Left Position',      description:
'This popover appears to the left of the target element with a rightward arrow.',    },  ]; rightPopoverData:
PopOverData[] = [    {      header: 'Right Position',      description:        'This popover appears to the right
of the target element with a leftward arrow.',    },  ]; startArrowPopoverData: PopOverData[] = [    {
header: 'Start Arrow',      description: 'Arrow positioned at the beginning of the popover.',    },  ];
centerArrowPopoverData: PopOverData[] = [    {      header: 'Center Arrow',      description: 'Arrow centered on
the popover (default).',    },  ]; endArrowPopoverData: PopOverData[] = [    {      header: 'End Arrow',
description: 'Arrow positioned at the end of the popover.',    },  ];
```

Available Positions

- Top- Appears above the target element with downward arrow

- Bottom- Appears below the target element with upward arrow

- Left- Appears to the left of the target element with rightward arrow

- Right- Appears to the right of the target element with leftward arrow

Arrow Alignment

- Start- Arrow positioned at the beginning of the popover

- Center- Arrow centered on the popover (default)

- End- Arrow positioned at the end of the popover

API Reference

Inputs

```
Property | Type | Default | Description
config | PopOverConfig | - | Configuration object for popover behavior
data | PopOverData[] | [] | Array of content items to display
```

```
config
```

```
PopOverConfig
```

```
data
```

```
PopOverData[]
```

```
[]
```

PopOverConfig Interface

```
Property | Type | Default | Description
arrow | 'start' | 'center' | 'end' | null | null | Arrow alignment position
position | 'top' | 'bottom' | 'left' | 'right' | 'top' | Popover position relative to target
left | number | 0 | Target element left position
top | number | 0 | Target element top position
bottom | number | 0 | Target element bottom position
width | number | 0 | Target element width
height | number | 0 | Target element height
showButtons | boolean? | false | Whether to show navigation buttons
showPagination | boolean? | false | Whether to show page counter
showIcon | boolean? | false | Whether to show icon navigation
showSkip | boolean? | false | Whether to show skip button
showLearnMore | boolean? | false | Whether to show learn more button
```

```
arrow
```

```
'start' | 'center' | 'end' | null
```

```
null
```

```
position
```

```
'top' | 'bottom' | 'left' | 'right'
```

```
'top'
```

```
left
```

```
number
```

```
0
```

```
top
```

number	
0	
bottom	
number	
0	
width	
number	
0	
height	
number	
0	
showButtons	
boolean?	
false	
showPagination	
boolean?	
false	
showIcon	
boolean?	
false	
showSkip	
boolean?	
false	
showLearnMore	
boolean?	
false	

PopOverData Interface

```
Property | Type | Default | Description
header | string | - | Header text for the popover
description | string | - | Description text for the popover
learnMoreUrl | string? | - | Optional URL for learn more functionality
```

header

string

description

string

learnMoreUrl

string?

Properties

```
Property | Type | Description
visible | boolean | Whether the popover is currently visible
currentIndex | number | Current page index for multi-page content
currentData | PopOverData | null | Current content data being displayed
currentPage | string | Current page display string (e.g., "2/5")
hasPrevious | boolean | Whether previous page is available
hasNext | boolean | Whether next page is available
```

visible

boolean

currentIndex

number

currentData

PopOverData | null

currentPage

string

hasPrevious

boolean

hasNext

boolean

Methods

```
Method | Parameters | Return Type | Description
```

```
previous() | event?: any | void | Navigate to previous page
next() | event?: any | void | Navigate to next page
skip() | event?: any | void | Close popover and skip content
learnMore() | event?: any | void | Open learn more URL in new tab
updatePosition() | - | void | Update popover position and arrow placement
setArrowPosition() | - | void | Set CSS custom properties for arrow positioning
```

```
previous()
```

```
event?: any
```

```
void
```

```
next()
```

```
event?: any
```

```
void
```

```
skip()
```

```
event?: any
```

```
void
```

```
learnMore()
```

```
event?: any
```

```
void
```

```
updatePosition()
```

```
void
```

```
setArrowPosition()
```

```
void
```

CSS Custom Properties

The component uses CSS custom properties for theming and styling:

```
Property | Default | Description
--popover-background | Dynamic | Background color of the popover
--popover-text | Dynamic | Text color of the popover
--popover-font | Dynamic | Font family for the popover
--popover-padding | Dynamic | Internal padding of the popover
--popover-border-radius | Dynamic | Border radius of the popover
--popover-max-width | Dynamic | Maximum width of the popover
--popover-z-index | Dynamic | Z-index for layering
--popover-shadow | Dynamic | Box shadow for the popover
--popover-border-color | Dynamic | Border color of the popover
--popover-content-gap | Dynamic | Gap between content sections
--popover-header-font-family | Dynamic | Font family for the header
--popover-header-font-size | Dynamic | Font size for the header
--popover-header-font-weight | Dynamic | Font weight for the header
--popover-header-color | Dynamic | Color for the header text
--popover-header-line-height | Dynamic | Line height for the header
--popover-description-font-family | Dynamic | Font family for the description
--popover-description-font-size | Dynamic | Font size for the description
--popover-description-font-weight | Dynamic | Font weight for the description
```

```
--popover-description-color | Dynamic | Color for the description text
--popover-description-line-height | Dynamic | Line height for the description
--popover-footer-margin-top | Dynamic | Top margin for the footer
--popover-pagination-font-size | Dynamic | Font size for pagination text
--popover-pagination-font-weight | Dynamic | Font weight for pagination text
--popover-pagination-color | Dynamic | Color for pagination text
--popover-pagination-font-family | Dynamic | Font family for pagination text
--popover-pagination-line-height | Dynamic | Line height for pagination text
--popover-button-gap | Dynamic | Gap between navigation buttons
--popover-button-width | Dynamic | Width of navigation buttons
--popover-button-height | Dynamic | Height of navigation buttons
--popover-disabled-opacity | Dynamic | Opacity for disabled button states
--popover-hover-background | Dynamic | Background color for button hover states
--popover-hover-border-radius | Dynamic | Border radius for button hover states
--popover-arrow-border-width | Dynamic | Width of arrow border
--popover-arrow-border-width-inner | Dynamic | Width of inner arrow border
--popover-arrow-border-shadow | Dynamic | Color of arrow border shadow
--popover-arrow-color | Dynamic | Color of arrow fill
```

--popover-background

--popover-text

--popover-font

--popover-padding

--popover-border-radius

--popover-max-width

--popover-z-index

--popover-shadow

--popover-border-color

--popover-content-gap

--popover-header-font-family

--popover-header-font-size

--popover-header-font-weight

--popover-header-color

--popover-header-line-height

--popover-description-font-family

--popover-description-font-size

--popover-description-font-weight

--popover-description-color

--popover-description-line-height

```
--popover-footer-margin-top
```

```
--popover-pagination-font-size
```

```
--popover-pagination-font-weight
```

```
--popover-pagination-color
```

```
--popover-pagination-font-family
```

```
--popover-pagination-line-height
```

```
--popover-button-gap
```

```
--popover-button-width
```

```
--popover-button-height
```

```
--popover-disabled-opacity
```

```
--popover-hover-background
```

```
--popover-hover-border-radius
```

```
--popover-arrow-border-width
```

```
--popover-arrow-border-width-inner
```

```
--popover-arrow-border-shadow
```

```
--popover-arrow-color
```

Best Practices

Content Guidelines

- Keep it concise: Popovers work best with short, focused content
- Use clear headers: Make the purpose of the popover immediately clear
- Provide actionable content: Include learn more links or navigation when appropriate
- Consider context: Position popovers to avoid covering important interface elements

Positioning Guidelines

- Choose appropriate positions: Use top/bottom for horizontal content, left/right for vertical content
- Align arrows thoughtfully: Center arrows for balanced content, start/end for directional emphasis
- Avoid boundaries: Ensure popovers don't extend beyond viewport edges
- Consider scrolling: Position popovers to remain visible during page scroll

Navigation Guidelines

- Match user expectations: Use familiar navigation patterns (prev/next, pagination)
- Provide clear feedback: Show current position and available actions

- Include skip options: Allow users to bypass content when appropriate

- Use consistent styling: Maintain visual consistency across navigation elements

Technical Notes

Component Architecture

The popover component uses a modern Angular architecture with:

- PopOverComponent- Main component with positioning and navigation logic

- IconComponentintegration for icon-based navigation

- ViewChild referencesfor DOM manipulation and positioning

- ChangeDetectionStrategy.OnPushfor performance optimization

Positioning System

The component implements an advanced positioning system:

- Automatic positioningbased on target element coordinates

- Arrow alignmentwith CSS custom properties for precise placement

- Boundary detectionto prevent overflow beyond viewport

- Scroll and resize handlingfor responsive positioning

Animation System

The component provides smooth entrance animations:

- Stretch effectsfor natural expansion from target elements

- Position-based animationswith appropriate transform origins

- CSS keyframesfor consistent and performant animations

- Smooth transitionsfor opacity and transform changes

Event Handling

The component manages multiple event types:

- Navigation eventsfor page changes and content progression

- User interactionfor skip, learn more, and navigation actions

- Window eventsfor scroll and resize handling

- Lifecycle eventsfor proper cleanup and memory management

CSS Architecture

The component uses a comprehensive CSS architecture:

- CSS custom propertiesfor theming and customization

- BEM-like namingfor maintainable and scalable styles

- Position-based modifiersfor different placement variants

- Responsive designwith flexible layouts and sizing

Accessibility Features

The component includes built-in accessibility features:

- Keyboard navigationfor all interactive elements

- ARIA attributesfor screen reader compatibility

- Focus managementfor proper tab order and focus indicators

- Semantic markupfor clear content structure and meaning