

Component Documentation

Header

The Header component is a flexible, responsive layout component designed to create consistent application headers and navigation bars. It features a three-column layout system with content projection, multiple theme variants, and intelligent responsive behavior that adapts to different screen sizes.

The component provides a structured approach to building headers with left, center, and right content sections, making it ideal for logos, navigation menus, search bars, user actions, and other header elements.

How to use

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

Basic Usage

Simple header with logo, navigation, and user actions.

```
<aava-header containerClass="">
  <div left-content class="header__left">
    <div class="logo"></div>
  </div>

  <div center-content class="header__center tabs-header">
    <aava-tabs
      [tabs]="iconsTabs"
      [activeTabId]="activeTabId"
      variant="default"
      size="sm"
      [showContentPanels]="false"
      (tabChange)="onTabChange($event)"
      ariaLabel="Basic navigation tabs"
    ></aava-tabs>
  </div>

  <div right-content class="header__right">
    <div class="search-box">
      <aava-icon iconName="search" [iconSize]="16" iconColor="#000"></aava-icon>
      <input
        type="text"
        [(ngModel)]="searchText"
        (input)="onSearchChange()"
        placeholder="Search"
      />
    </div>

    <aava-icon
      class="icon"
      iconName="Sun"
      [iconSize]="20"
      [iconColor]="getIconColor"
      (click)="onGraphIconClick()"
    ></aava-icon>
    <aava-icon
      class="icon"
      iconName="Languages"
      [iconSize]="20"
      [iconColor]="getIconColor"
      (click)="onTranslateIconClick()"
    ></aava-icon>
    <aava-icon
      class="icon"
      iconName="EllipsisVertical"
      [iconSize]="20"
      [iconColor]="getIconColor"
      (click)="onDayLightModeIconClick()"
    ></aava-icon>

    <aava-avatars
      size="md"
      shape="square"
      [imageUrl]="getLoggedInUserLogo()"
      badgeState="high-priority"
      badgeSize="lg"
      [badgeCount]="9"
    ></aava-avatars>
  </div>
</aava-header>
```

```
logoUrl: string = "assets/asc-logo.png";
activeTabId: string = 'overview';

private baseIconsTabs: DemoTabItem[] = [
  {
    id: 'tab1',
    label: 'Tab1',
    content: 'Content 1',
    iconName: 'chevron-right',
  },
  {
    id: 'tab2',
    label: 'Tab2',
    content: 'Content 2',
    iconName: 'chevron-right',
  },
  {
    id: 'tab3',
    label: 'Tab3',
    content: 'Content 3',
    iconName: 'chevron-right',
  },
  {
    id: 'tab4',
    label: 'Tab4',
    content: 'Content 4',
    iconName: 'chevron-right',
  },
];
accounts = [
  {
    companyName: 'ASCENDION',
    email: 'Jhon.Stark@Ascendion.com',
    companyLogo: 'assets/logos/dlogo.svg',
    userLogo: 'assets/1.svg',
    isLoggedIn: true,
  },
  {
    companyName: 'Axos',
    email: 'Akash.Kumar@Axos.com',
    companyLogo: 'assets/logos/dlogo.svg',
    userLogo: 'assets/1.svg',
    isLoggedIn: false,
  },
];
get iconsTabs(): TabItem[] {
  return this.baseIconsTabs.map((tab) => ({
    ...tab,
    active: tab.id === this.activeTabId,
  }));
}

get getIconColor(): string {
  return '#000';
}

onTabChange(event: any) {
```

```
    this.activeTabId = event.id;
}

onSearchChange() {
  console.log('Search text changed:', this.searchText);
}

onGraphIconClick() {
  console.log('Graph Icon Clicked');
}

onTranslateIconClick() {
  console.log('Translate Icon Clicked');
}

onDayLightModeIconClick() {
  console.log('Day Light Mode Icon Clicked');
}

getLoggedInUserLogo(): string {
  const loggedInAccount = this.accounts.find((account) => account.isLoggedIn);
  return loggedInAccount ? loggedInAccount.userLogo : 'assets/1.svg';
}
```

Variations

Without Navigation

```

<aava-header containerClass="">
  <div left-content class="header_left">
    <div class="logo"></div>
  </div>

  <div right-content class="header_right">
    <div class="search-box">
      <aava-icon iconName="search" [iconSize]="16" iconColor="#000"></aava-icon>
      <input
        type="text"
        [(ngModel)]="searchText"
        (input)="onSearchChange()"
        placeholder="Search"
      />
    </div>

    <aava-icon
      class="icon"
      iconName="Sun"
      [iconSize]="20"
      [iconColor]="getIconColor"
      (click)="onGraphIconClick()"
    ></aava-icon>

    <aava-icon
      class="icon"
      iconName="Languages"
      [iconSize]="20"
      [iconColor]="getIconColor"
      (click)="onTranslateIconClick()"
    ></aava-icon>

    <aava-icon
      class="icon"
      iconName="EllipsisVertical"
      [iconSize]="20"
      [iconColor]="getIconColor"
      (click)="onDayLightModeIconClick()"
    ></aava-icon>

    <aava-avatars
      size="md"
      shape="square"
      [imageUrl]="getLoggedInUserLogo()"
      badgeState="low-priority"
      badgeSize="lg"
      [badgeCount]="9"
    ></aava-avatars>
  </div>
</aava-header>

---

logoUrl: string = "assets/asc-logo.png";

accounts = [
  {
    companyName: 'ASCENDION',
    email: 'Jhon.Stark@Ascendion.com',
    companyLogo: 'assets/logos/dlogo.svg',

```

```

    userLogo: 'assets/1.svg',
    isLoggedIn: true,
  },
  {
    companyName: 'Axos',
    email: 'Akash.Kumar@Axos.com',
    companyLogo: 'assets/logos/dlogo.svg',
    userLogo: 'assets/1.svg',
    isLoggedIn: false,
  },
];
}

get iconsTabs(): TabItem[] {
  return this.baseIconsTabs.map((tab) => ({
    ...tab,
    active: tab.id === this.activeTabId,
  }));
}

get getIconColor(): string {
  return '#000';
}

onSearchChange() {
  console.log('Search text changed:', this.searchText);
}

onGraphIconClick() {
  console.log('Graph Icon Clicked');
}

onTranslateIconClick() {
  console.log('Translate Icon Clicked');
}

onDayLightModeIconClick() {
  console.log('Day Light Mode Icon Clicked');
}

getLoggedInUserLogo(): string {
  const loggedInAccount = this.accounts.find((account) => account.isLoggedIn);
  return loggedInAccount ? loggedInAccount.userLogo : 'assets/1.svg';
}

```

A compact header layout that omits the navigation section—ideal for minimal pages or dashboards.

Without Search

```

<aava-header containerClass="">
  <div left-content class="header__left">
    <div class="logo"></div>
  </div>

  <div center-content class="header__center tabs-header">
    <aava-tabs
      [tabs]="iconsTabs"
      [activeTabId]="activeTabId"
      variant="default"
      size="sm"
      [showContentPanels]="false"
      (tabChange)="onTabChange($event)"
      ariaLabel="Basic navigation tabs"
    ></aava-tabs>
  </div>

  <div right-content class="header__right">
    <aava-icon
      class="icon"
      iconName="Sun"
      [iconSize]="20"
      [iconColor]="getIconColor"
      (click)="onGraphIconClick()"
    ></aava-icon>

    <aava-icon
      class="icon"
      iconName="Languages"
      [iconSize]="20"
      [iconColor]="getIconColor"
      (click)="onTranslateIconClick()"
    ></aava-icon>

    <aava-icon
      class="icon"
      iconName="EllipsisVertical"
      [iconSize]="20"
      [iconColor]="getIconColor"
      (click)="onDayLightModeIconClick()"
    ></aava-icon>

    <aava-avatars
      size="md"
      shape="square"
      [imageUrl]="getLoggedInUserLogo()"
      badgeState="medium-priority"
      badgeSize="lg"
      [badgeCount]="9"
    ></aava-avatars>
  </div>
</aava-header>

```

```

---
logoUrl: string = "assets/asc-logo.png";
activeTabId: string = 'overview';

private baseIconsTabs: DemoTabItem[] = [
  {

```

```

    id: 'tab1',
    label: 'Tab1',
    content: 'Content 1',
    iconName: 'chevron-right',
},
{
  id: 'tab2',
  label: 'Tab2',
  content: 'Content 2',
  iconName: 'chevron-right',
},
{
  id: 'tab3',
  label: 'Tab3',
  content: 'Content 3',
  iconName: 'chevron-right',
},
{
  id: 'tab4',
  label: 'Tab4',
  content: 'Content 4',
  iconName: 'chevron-right',
},
];
accounts = [
{
  companyName: 'ASCENDION',
  email: 'Jhon.Stark@Ascendion.com',
  companyLogo: 'assets/logos/dlogo.svg',
  userLogo: 'assets/1.svg',
  isLoggedIn: true,
},
{
  companyName: 'Axos',
  email: 'Akash.Kumar@Axos.com',
  companyLogo: 'assets/logos/dlogo.svg',
  userLogo: 'assets/1.svg',
  isLoggedIn: false,
},
];
get iconsTabs(): TabItem[] {
  return this.baseIconsTabs.map((tab) => ({
    ...tab,
    active: tab.id === this.activeTabId,
  }));
}

get getIconColor(): string {
  return '#000';
}

onTabChange(event: any) {
  this.activeTabId = event.id;
}

onGraphIconClick() {
  console.log('Graph Icon Clicked');
}

```

```
onTranslateIconClick() {
  console.log('Translate Icon Clicked');
}

onDayLightModeIconClick() {
  console.log('Day Light Mode Icon Clicked');
}

getLoggedInUserLogo(): string {
  const loggedInAccount = this.accounts.find((account) => account.isLoggedIn);
  return loggedInAccount ? loggedInAccount.userLogo : 'assets/l.svg';
}
```

Displays a clean header without the search bar—suitable for content-focused layouts or authenticated user dashboards.

Mobile View

```
<aava-header containerClass="">
  <div left-content class="header__left">
    <aava-icon
      class="menu-toggle"
      iconName="Menu"
      [iconSize]="24"
      iconColor="#000"
      (click)="toggleMenuDropdown()"
    ></aava-icon>

    <div class="menu-dropdown" [class.open]="isMenuOpen">
      <div class="dropdown-logo">
        
      </div>

      <div class="search-box dropdown-search">
        <aava-icon
          iconName="search"
          [iconSize]="16"
          iconColor="#000"
        ></aava-icon>
        <input
          type="text"
          [(ngModel)]="searchText"
          (input)="onSearchChange()"
          placeholder="Search"
        />
      </div>

      <div class="dropdown-tabs">
        <aava-tabs
          [tabs]="iconsTabs"
          [activeTabId]="activeTabId"
          variant="default"
          size="sm"
          [showContentPanels]="false"
          orientation="vertical"
          (tabChange)="onTabChange($event)"
        >
        </aava-tabs>
      </div>
    </div>
  </div>

  <div right-content class="header__right">
    <aava-icon
      class="icon"
      iconName="Sun"
      [iconSize]="20"
      [iconColor]="getIconColor"
      (click)="onGraphIconClick()"
    ></aava-icon>

    <aava-icon
      class="icon"
      iconName="Languages"
      [iconSize]="20"
      [iconColor]="getIconColor"
      (click)="onTranslateIconClick()"
    ></aava-icon>
  </div>
</aava-header>
```

```

<aava-avatars
  size="md"
  shape="square"
  [imageUrl]="getLoggedInUserLogo()"
  badgeState="information"
  badgeSize="lg"
  [badgeCount]="9"
></aava-avatars>

<aava-icon
  class="icon"
  iconName="EllipsisVertical"
  [iconSize]="20"
  [iconColor]="getIconColor"
  (click)="onDayLightModeIconClick()"
></aava-icon>
</div>
</aava-header>

---

logoUrl: string = "assets/asc-logo.png";
searchText = "";
isDropdownOpen = true;
activeTabId: string = "overview";
isMenuOpen = false;

private baseIconsTabs: DemoTabItem[] = [
  {
    id: 'tab1',
    label: 'Tab1',
    content: 'Content 1',
    iconName: 'chevron-right',
  },
  {
    id: 'tab2',
    label: 'Tab2',
    content: 'Content 2',
    iconName: 'chevron-right',
  },
  {
    id: 'tab3',
    label: 'Tab3',
    content: 'Content 3',
    iconName: 'chevron-right',
  },
  {
    id: 'tab4',
    label: 'Tab4',
    content: 'Content 4',
    iconName: 'chevron-right',
  },
];
accounts = [
  {
    companyName: 'ASCENDION',
    email: 'Jhon.Stark@Ascendion.com',
    companyLogo: 'assets/logos/dlogo.svg',
    userLogo: 'assets/1.svg',
  }
];

```

```

        isLoggedIn: true,
    },
    {
        companyName: 'Axos',
        email: 'Akash.Kumar@Axos.com',
        companyLogo: 'assets/logos/dlogo.svg',
        userLogo: 'assets/1.svg',
        isLoggedIn: false,
    },
];
}

get getIconColor(): string {
    return '#000';
}

get iconsTabs(): TabItem[] {
    return this.baseIconsTabs.map((tab) => ({
        ...tab,
        active: tab.id === this.activeTabId,
    }));
}

onSearchChange() {
    console.log('Search text changed:', this.searchText);
}

toggleMenuDropdown() {
    this.isMenuOpen = !this.isMenuOpen;
}

onDayLightModeIconClick() {
    console.log('Day Light Mode Icon Clicked');
}

onTranslateIconClick() {
    console.log('Translate Icon Clicked');
}

onGraphIconClick() {
    console.log('Graph Icon Clicked');
}

onTabChange(event: any) {
    this.activeTabId = event.id;
}

getSortedAccounts() {
    return this.accounts.sort((a, b) =>
        a.isLoggedIn === b.isLoggedIn ? 0 : a.isLoggedIn ? -1 : 1
    );
}

getLoggedInUserLogo(): string {
    const loggedInAccount = this.accounts.find((account) => account.isLoggedIn);
    return loggedInAccount ? loggedInAccount.userLogo : 'assets/1.svg';
}

```

Responsive version optimized for smaller screens, with collapsed or hidden center content and accessible mobile actions.

Content Layouts

Layout Patterns

- Logo + Navigation + Actions - Traditional header with main navigation
- Logo + Search + User Menu - Application header with search functionality
- Minimal Logo + Navigation - Clean, simple header design

Responsive Behavior

Responsive Breakpoints

- Above 1440px : Full three-column layout
- Below 1440px : Center section with visual separator, right section pinned
- Below 1280px : Center section hidden, simplified mobile layout

Features

Three-Column Layout System

- Left Content : Typically for logos, brand elements, or primary navigation
- Center Content : Main navigation, search bars, or secondary content
- Right Content : User actions, notifications, or tertiary navigation

Content Projection

- Uses Angular content projection with left-content , center-content , and right-content selectors
- Flexible content placement without component restrictions
- Easy customization and styling

Theme System

- Light, dark, and transparent theme variants
- Consistent color schemes across themes
- Easy theme switching and customization

Responsive Design

- Automatic layout adjustments based on screen size
- Visual separators that appear/disappear appropriately
- Mobile-optimized layouts

Custom Styling

- containerClass input for additional CSS classes
- Flexible styling through content projection
- CSS custom properties for theme customization

API Reference

Input Properties

Property	Type	Default	Description
theme	'light' 'dark' 'transparent'	'light'	Header theme variant
containerClass	string	"	Additional CSS classes for the header container

Content Projection Selectors

Selector	Description
[left-content]	Left section content (logo, brand)
[center-content]	Center section content (navigation, search)
[right-content]	Right section content (actions, user menu)

Best Practices

Content Organization

- Left Section : Place logos, brand names, or primary navigation
- Center Section : Use for main navigation menus or search functionality
- Right Section : Include user actions, notifications, or secondary navigation

Responsive Considerations

- Design for mobile-first approach
- Ensure center content gracefully hides on smaller screens
- Test layout behavior across different breakpoints

Theme Usage

- Light Theme : Standard applications, content-heavy pages
- Dark Theme : Dashboard applications, media-focused interfaces
- Transparent Theme : Hero sections, overlay headers, landing pages

Accessibility

- Ensure sufficient color contrast in all themes
- Provide clear navigation structure
- Use semantic HTML elements within content sections

Styling

The header component uses CSS Grid and Flexbox for layout management with:

- Automatic spacing and alignment
- Responsive breakpoints with CSS media queries
- Theme-based color schemes
- Visual separators for center content
- Smooth transitions and animations