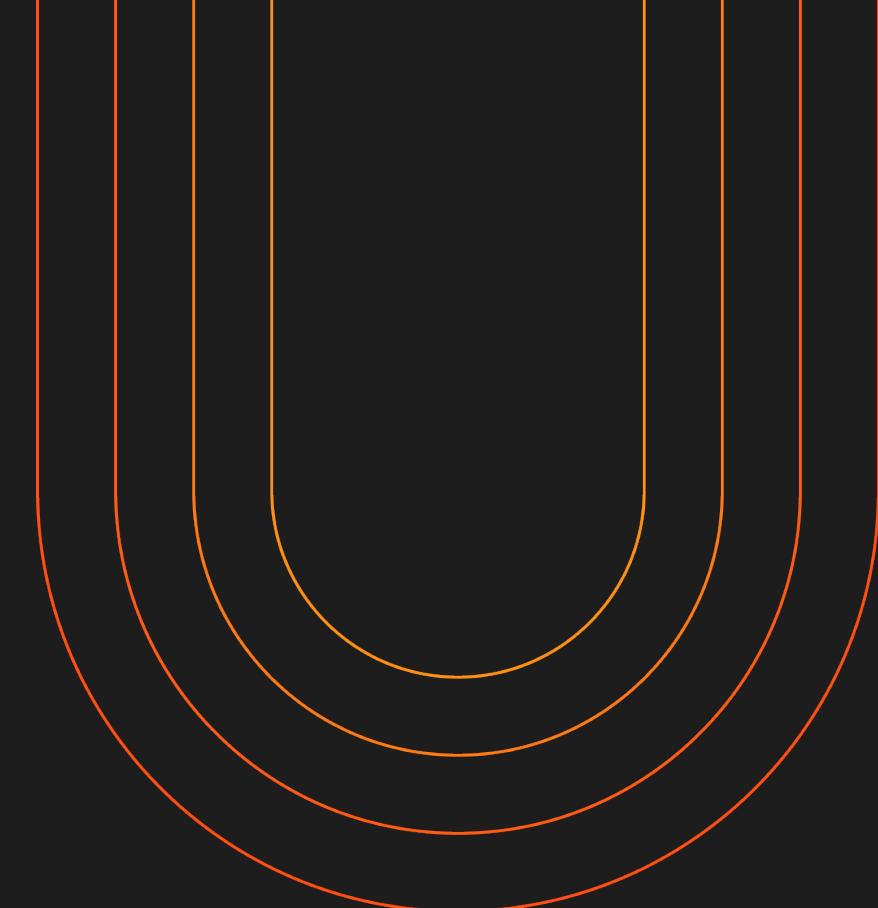


Boost your UI5 skills for creating accessibility apps

Dobrin Dimchev, SAP Labs Bulgaria
July 8th, 2022



Agenda

Accessibility Documentation

- Improvements in Demo Kit documentation
- End-user screen reader and keyboard handling documentation

Best Practice Application

- Accessibility-first samples

Accessibility Mission in SAP Discovery Center

- Learning accessibility in practice

Accessibility Documentation

- ▼ Essentials
 - ▼ Accessibility
 - OpenUI5 Accessibility Features
 - Accessibility Support History
- ▼ Developing Apps
 - ▼ Accessibility
 - Screen Reader Support
 - Keyboard Handling
 - Colors and Theming
 - Text Size and Fonts
 - Recommendations
- ▼ Developing Controls
 - ▼ Accessibility Aspects
 - Keyboard Handling for OpenUI5 Controls for Developers
 - Screen Reader Support for OpenUI5 Controls
 - Theming

Accessibility Documentation

Understanding Accessibility UI5 Accessibility Building Blocks

The screenshot shows the OpenUI5 documentation website with the URL [https://openui5.org/api/doc/api/#/topic/0f3a3a3a3a3a3a3a](#). The page title is "Accessibility". The left sidebar contains a navigation tree with various topics, and the "Accessibility" topic is highlighted with a blue border. The main content area starts with a section titled "Understanding Accessibility" which defines accessibility as the possibility for everyone to access and use information technology products. It then transitions to "OpenUI5 Accessibility Building Blocks", listing several key components: Markup (correct HTML), ARIA attribute, Keyboard handling, Screen reader, Contrast, Resizing, and APIs. Below this, there is a "Related information" section with links to "Accessibility in SAP Fiori", "Accessibility in the Developing Apps Section", and "Accessibility Aspects in the Developing Controls Section".

Accessibility Documentation

Accessibility Support History

- Enhancements
- Updates
- New accessibility concepts
- Important SAP Notes

The screenshot shows the 'Accessibility Support History' page of the OpenUI5 Documentation. At the top, there's a navigation bar with links for Documentation, API Reference, Samples, Demo Apps, and Tools. To the right of the navigation is the text 'Version 1.105.0 - development in progress.' and a 'Change Version' button. Below the navigation, the page title 'Accessibility Support History' is displayed. A note states that OpenUI5 follows SAP's design and development guidelines, based on WCAG 2.1 level A and AA. A table lists accessibility features and their availability versions:

Feature	Available as of version
HCW Theme	1.46
Screen Reader Support	1.30
Keyboard Handling	1.26
HCB Theme	1.26

Below this, a table provides details about enhancements and updates:

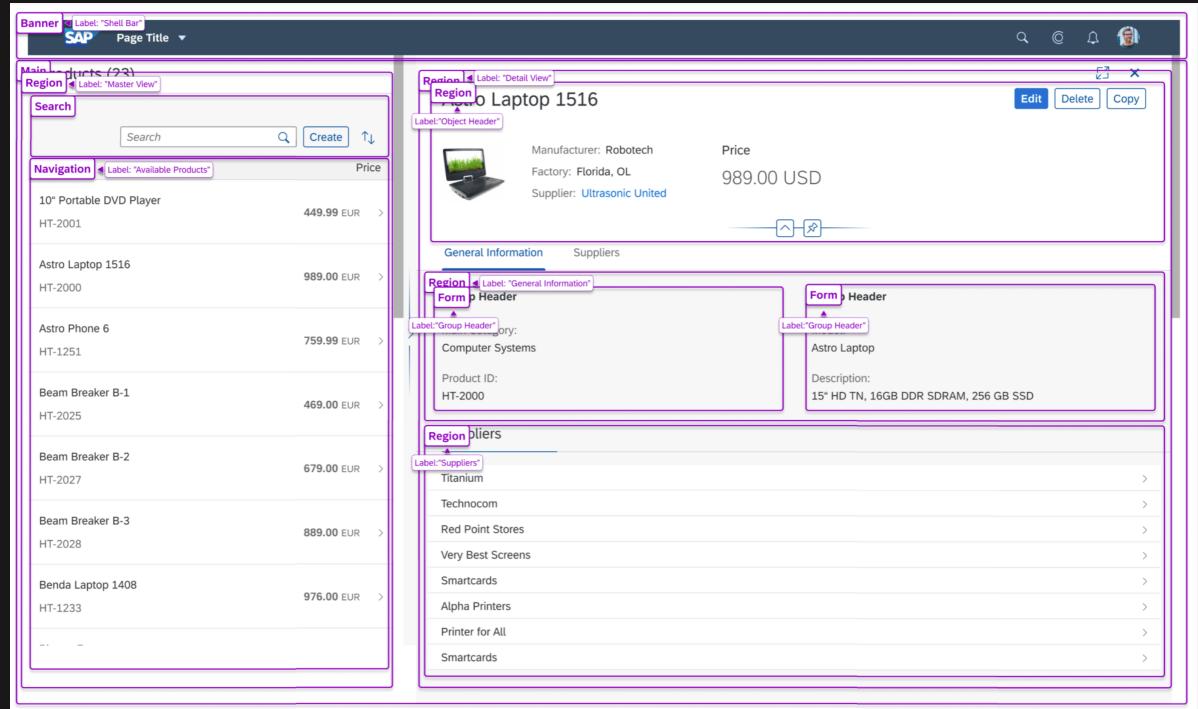
Enhancements and Updates	As of version
Update of the reference test environment. Latest update with UI5 1.102	For the versions, see SAP Note 2564165 .
We have prevented the overwrite of the Windows high contrast setting for all themes when using Chrome and Edge browsers.	1.101
We have enhanced OpenUI5 accessibility support according to the latest WAI-ARIA 1.1 specification	1.84
we have prevented the automatic insertion of role application on the body of the OpenUI5 applications. After the change, a mode-based screen reader will start operating in reading mode as its default mode. For more information, please refer to SAP Note 2925884 .	1.78
OpenUI5 is following the SAP's updated design and development guidelines, as well as the testing procedures and accessibility reporting, that are based on WCAG 2.1 level A and AA.	1.75

Accessibility Documentation

Developing Apps

- Logical groups
- New recommendations

- ▼ Developing Apps
 - ▼ Accessibility
 - Screen Reader Support
 - Landmark API
 - Labeling and Tooltips
 - Invisible Messaging
 - Dialogs, Popups, and Popovers
 - Keyboard Handling
 - Fast Navigation
 - Colors and Theming
 - Text Size and Fonts
 - Recommendations



Accessibility Best Practice Application

Accessibility Guide

The Accessibility Guide provides in-depth information with examples about web accessibility, whether you are a control or application developer, or a business user searching to learn more about web accessibility in the SAPUI5 area.

Accessibility Guide

Accessibility at Application Level

Many requirements are already covered by the technology framework, but design aspects that are related to the individual purpose of the application still need to be implemented by the individual design teams. Here are some examples:

Keyboard Operation

- Initial focus position: To enable users to work through a task efficiently, always set an initial focus for the task. Set the focus to either the first logical interaction element or the first element in the task.
- Structure and fast navigation: Ensure that the navigation within your application is logical and follows the task structure for the intended purpose. When designing apps with a large set of functions or information blocks, always form logical groups to help users navigate efficiently. You can form logical groups using a container, a toolbar, or other grouping elements. To enable faster navigation, SAPUI5 allows users to skip groups (with F6 or Shift + F6). Also, enable direct keyboard navigation to logical groups, such as working areas or navigation areas.

Accessibility Best Practice Application

Located in Demo Kit Tools Page.

Provides **guidance** for developers to watch and follow best practices when developing accessibility features.

The screenshot shows the SAPUI5 Accessibility Guide application interface. The top navigation bar includes icons for home, accessibility guide, overview, and explore. The left sidebar has a tree view with nodes like 'Introduction' (selected), 'Application Developer' (expanded), 'Labeling and Description', 'Landmark API', 'Focus Handling', 'Invisible Content', 'Empty Display Control Re...', 'Message Handling', 'Control Developer' (expanded), 'Screen-Reader Support', and 'Keyboard-Handling Support'. The main content area is titled 'Introduction' and 'Motivation'. It contains a message box stating 'This application is live and still evolving, we welcome your contribution and feedback.' Below it, text explains the purpose of the application for developers and business users. It then defines accessibility as the possibility for everyone, including people with disabilities, to access and use information technology products. The application aims to bring accessibility to life by providing appropriate features and following SAP's design and development guidelines, based on WCAG 2.1 level A and AA. It highlights SAP's investment in accessibility and its incorporation into the framework and application levels. Finally, it mentions the complex process of enabling accessibility and provides related links to SAP Software Accessibility and Accessibility in Demokit.

This application is live and still evolving, we welcome your contribution and feedback.

In this application you will understand what web accessibility is and how you can implement it. Whether you are a control or application developer, or a business user searching to know more about accessibility in the SAPUI5 area, you can find a lot of useful information and links in this app.

But first let's start with the definition of accessibility.

Accessibility refers to the possibility for everyone, including and especially people with disabilities, to access and use information technology products. SAP has long made this a priority. Making our software work for more people has been and continues to be a high priority. All users should be able to operate our software without loss of meaningful content, functionality, and efficiency, by using assistive technologies.

Following those principles in an ongoing approach, we at SAPUI5 aim to bring accessibility to life by providing the appropriate accessibility features, and following precise accessibility requirements and processes. SAPUI5 is following the SAP's design and development guidelines, as well as the testing procedures and accessibility reporting, that are based on Web Content Accessibility Guidelines (WCAG) 2.1 level A and AA.

We are heavily invested in accessibility and want to show you how by using the framework you can achieve an accessible application.

Accessibility in SAPUI5 is incorporated in two levels: framework and application. Many fundamental accessibility features are built into the core design elements upfront and are available out of the box.

Enabling accessibility is a complex process and in this application, we will guide you through the most important steps.

Related Links:

- [SAP Software Accessibility](#)
- [Accessibility in Demokit](#)

Accessibility Best Practice Application

Complete code examples

Semantically organized content

The screenshot shows the SAP UI5 code editor interface. On the left is a sidebar with a navigation tree. The 'Labeling and Description' section is currently selected. The main area displays four files: View.view.xml, index.html, manifest.json, and Component.js. The View.view.xml file contains XML code for a View component, including an App and Page structure with various attributes like id, backgroundDesign, and landmarkInfo. The index.html file shows the URL 'index.html'. The manifest.json file contains the application's manifest configuration. The Component.js file shows the controller logic.

The screenshot shows the SAP Accessibility Guide interface. The 'Select' component is selected in the sidebar. The main content area shows two examples: 'Using Label and labelFor property:' and 'Using InvisibleText and ariaLabelledBy:'. The 'Controller.controller.js' file shows the controller logic for the Select component, including the use of 'labelFor' and 'ariaLabelledBy' properties. The 'View.view.xml' file shows the XML structure, including the 'label' element with 'labelFor' and the 'Select' element with 'ariaLabelledBy'. The 'index.html' file shows the rendered UI.

Accessibility Best Practice Application

More to come

Live editor

More contributions

The screenshot shows a web application titled "Accessibility Guide". The header includes a menu icon, a logo with a flame, and three navigation links: "Overview" (underlined), "Explore", and "Introduction". A sidebar on the left contains a "Navigation" section with "Introduction" and "Application Developer" (with a dropdown arrow), followed by five "Content" items: "Labeling and Description", "Landmark API", "Focus Handling", and "Invisible Content". The main content area is titled "Introduction" and features a large heading "Motivation". At the bottom of the main content area is a callout box containing the text: "This application is live and still evolving, we welcome your contribution and feedback."

Accessibility Mission in SAP Discovery Center

Accessibility mission in SAP Discovery Center

Bring more visibility about accessibility in UI5 and more easily and practically get to know the basic accessibility features and APIs that are offered by the framework.

The screenshot shows the SAP Discovery Center interface with the following details:

- Title:** Accessibility mission in SAP Discovery Center
- Started by:** Dobrin Dimchev
- Started on:** Nov 24, 2021
- Progress:** 0%
- Mission Phase:** Discover
- Project Board Tabs:** Overview, **Project Board**, Resources, Support, Related Missions
- Search Bar:** Search by Name, Members and Labels
- Filters:** Filters icon
- Discover Phase (Optional):**
 - Accessibility - Free Everyone's Potential
- Discover Phase (Important, Basics):**
 - SAPUI5 Accessibility Features
- Discover Phase (Optional):**
 - Assistive Technologies
- Discover Phase (Important, Must):**
 - Prerequisites
- Prepare Phase:**
 - Prepare your SAP Business Application Studio
 - Get familiar with the code
- Add a card:** + Add a card
- Develop I Phase (Exercise):**
 - Panel with inputs and labels
- Develop I Phase (Exercise):**
 - Simple Form
- Develop I Phase (Exercise):**
 - Simple Form Validation
- Develop I Phase (Exercise):**
 - Simple Form Validation - Part 2: Add type and required

Accessibility mission in SAP Discovery Center

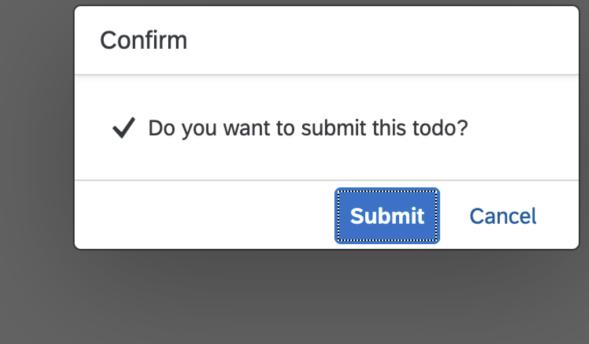
Develop

- Labeling
- Forms
- Validation
- Dialogs
- Messaging
- Landmark

Dialogs

Introduction

Dialogs are common way to provide information to the user after he interacts with them. That's why it is important to be fully accessible as very often the selections in them



Exercise

Panel with inputs and labels

Introduction

In the Todo application, users should first fill a simple form containing inputs and labels.

In this exercise, you are going to learn how to connect the Label controls with the Input controls manually. These examples are based on the `Panel` control and since it is a generic container the labelling functionality should be done manually. Later in this mission you will learn how to use the Form-based approach, and benefit from its features to automatically label the inputs.

Exercise

Initial setup

- Have the application served using the steps from the first exercise.
- Start Jaws screen reader.
- Explore the content of the Panel audible.

Connect the labels

- Problem: As you may have already noticed, in the initial test with Jaws, the labels of the fields are not read out. Your task is to connect the labels with the relevant input fields.
- Acceptance criteria: Jaws announces the labels of the input fields when they are focused.
- Hints:
 - Enhance the code in the `Todo.view.xml` file.
 - Read more about the `Label` control and check the available associations on this page: <https://ui5.sap.com/#/api/sap.m.Label%23associations>
 - Use the `labelFor association`. You should pass the ID of the labelled input. For the `CheckBox` you have two possibilities. The first one is to connect it with `labelFor` like it is described above. The other option is to use the `text` property of the `CheckBox` and the labelling will be done internally. You can read more about the properties here: <https://ui5.sap.com/#/api/sap.m.CheckBox%23controlProperties>.
 - Since `labelFor` accepts only one control, for the name inputs of the assignee, try to use the `ariaLabelledBy` association. Read more about the `InvisibleText` control: <https://ui5.sap.com/#/api/sap.ui.core.InvisibleText>.
 - Try to create two instances of the control with the relevant texts ("First Name" and "Last Name").
 - Reference the created controls via the `ariaLabelledBy` association of the inputs. You should pass the IDs of the newly created `InvisibleTexts`.
 - Validate the panel again with JAWS.

Key Takeaways & Resources

Accessibility Documentation

- <https://ui5.sap.com/#/topic/322f55d0cf1e4b459cc1911c899b7a5f>
- Essentials, Developing apps and controls
- Enhancements and updates
- Refers to best practice samples

Accessibility Best Practice Application

- <https://ui5.sap.com/test-resources/sap/m/demokit/accessibilityGuide/webapp/index.html>
- Complete accessibility-first samples

Accessibility Mission

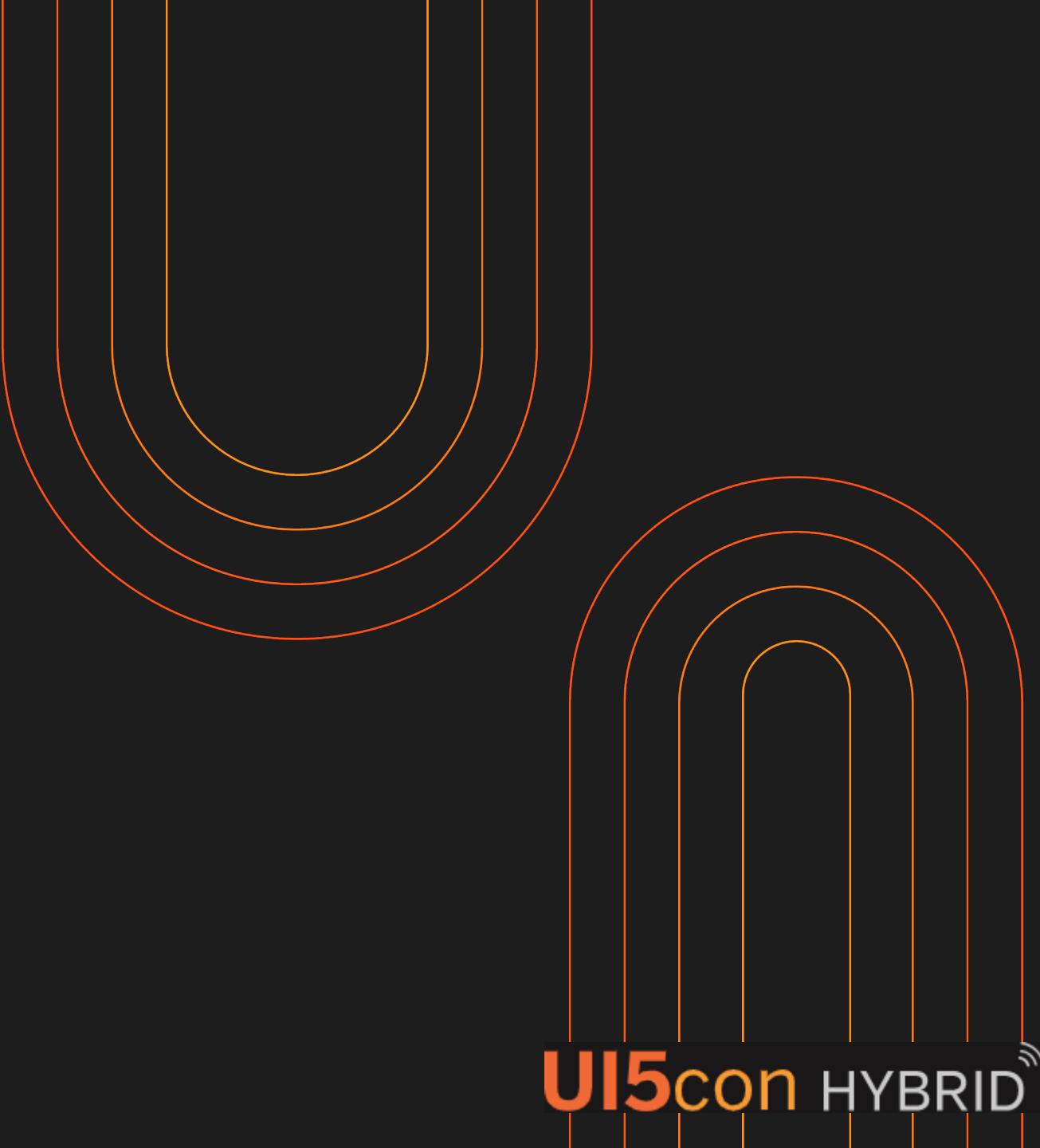
- <https://discovery-center.cloud.sap/missiondetail/3530/3571/>
- Make your application more accessible to all users

Thank you!

Dobrin Dimchev

JavaScript Developer

dobrin.dimchev@sap.com



UI5con HYBRID