

# Tizen Web Application with TAU



\* This document is based on Tizen Studio

# Table of Contents

Overview of TAU	3
What is TAU?	4
TAU in Web Application Development Process	5
Core Concepts of TAU	12
Role of TAU	13
Getting Started with TAU	14
Creating Simple UI with TAU	20
Page Structure	21
Page	22
Listview	23

# Table of Contents

## Creating Simple UI with TAU

SnapListview	24
Page Navigation	27
Checkbox	29
Progress	30
Appendix	32
TAU UI Components	33
TAU UI Components Sample Application	36
TAU References	38

# Overview of TAU

# What is TAU?

TAU stands for Tizen Advanced UI framework, which is the Tizen Web UI framework.

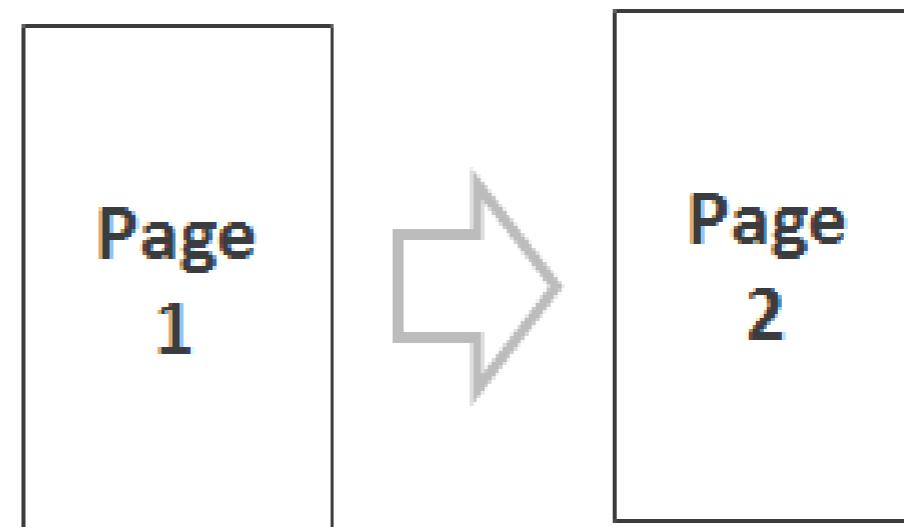
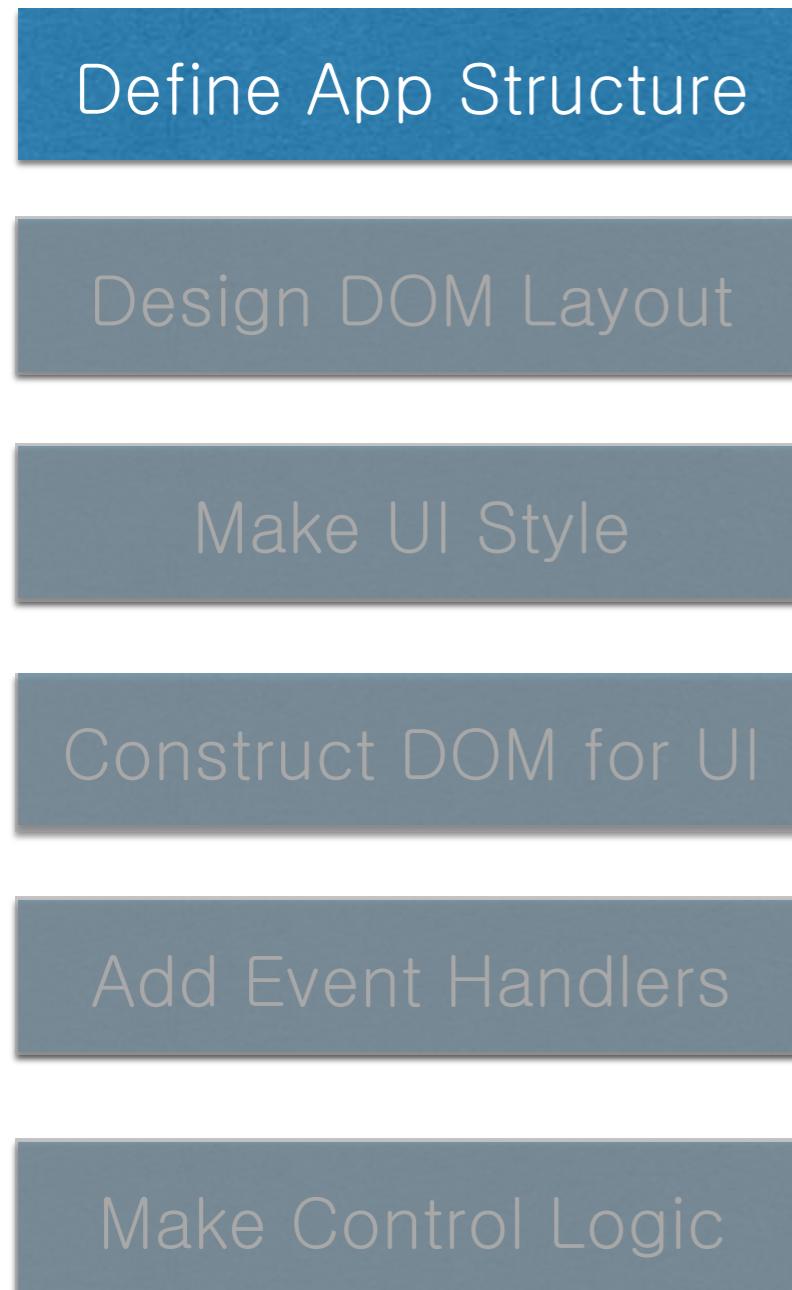
TAU helps developers to:

- Create and manage pages and various kinds of UI components
- Use utilities for Web applications such as using transition effect for pages.
- Support the mobile and wearable profiles.
- Support the circular UI for wearable devices.



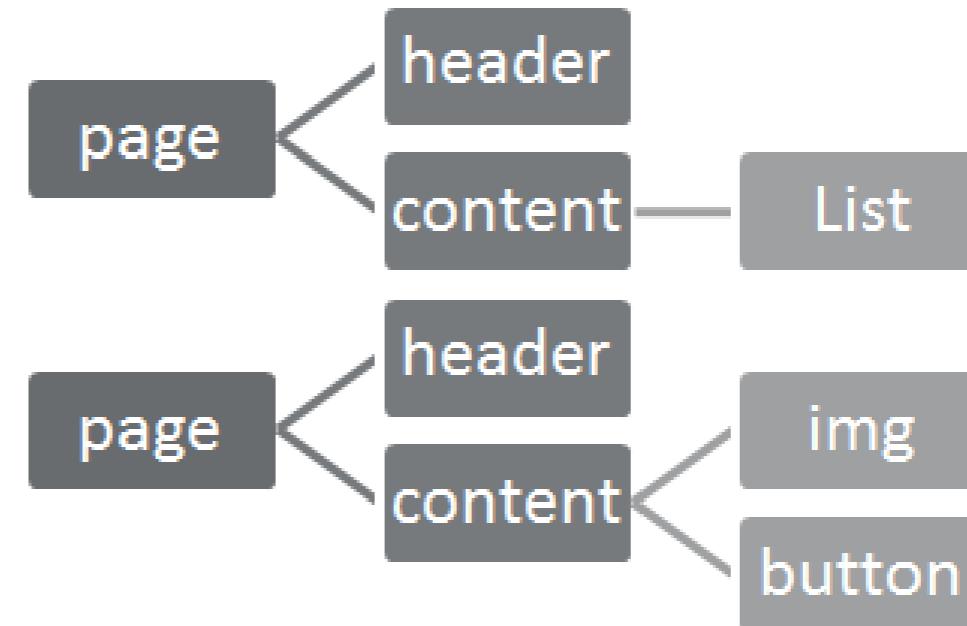
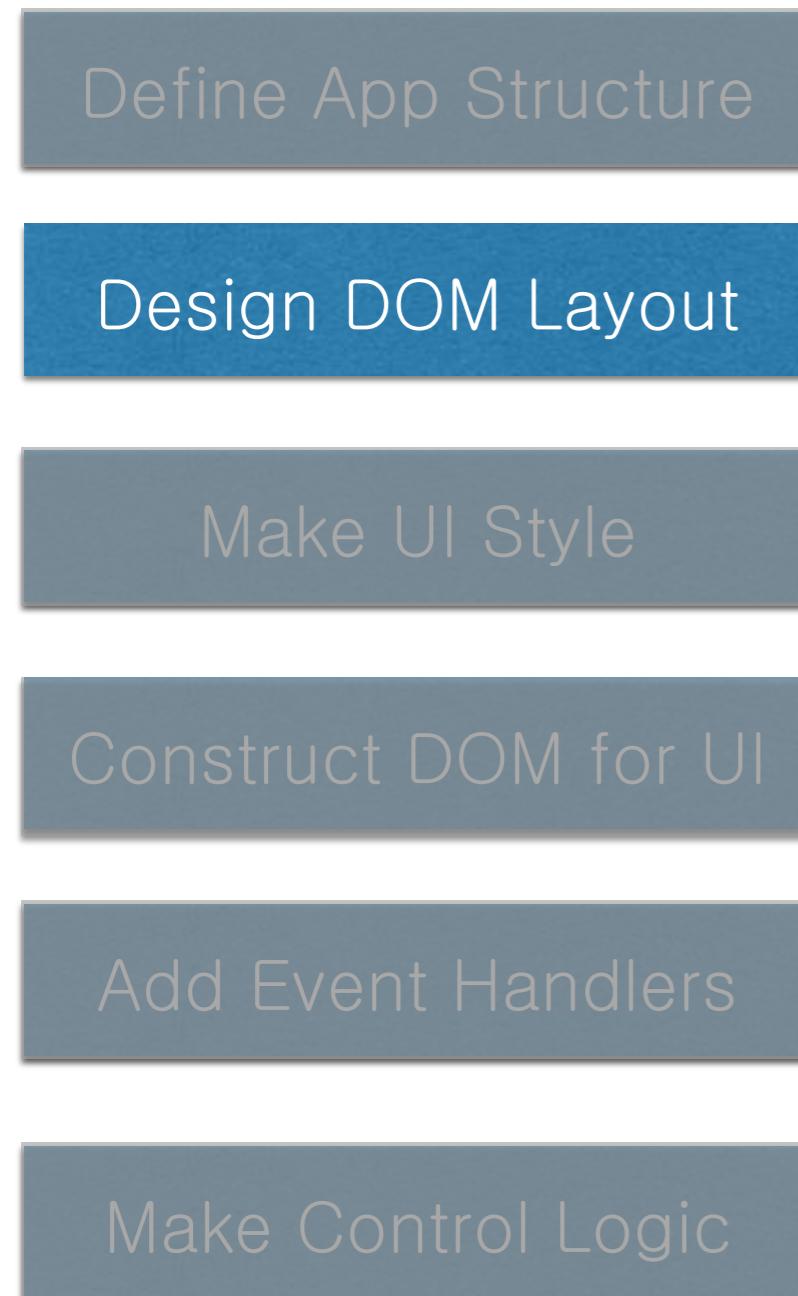
# TAU in Web Application Development Process

The following figure shows a typical workflow of Web application development.



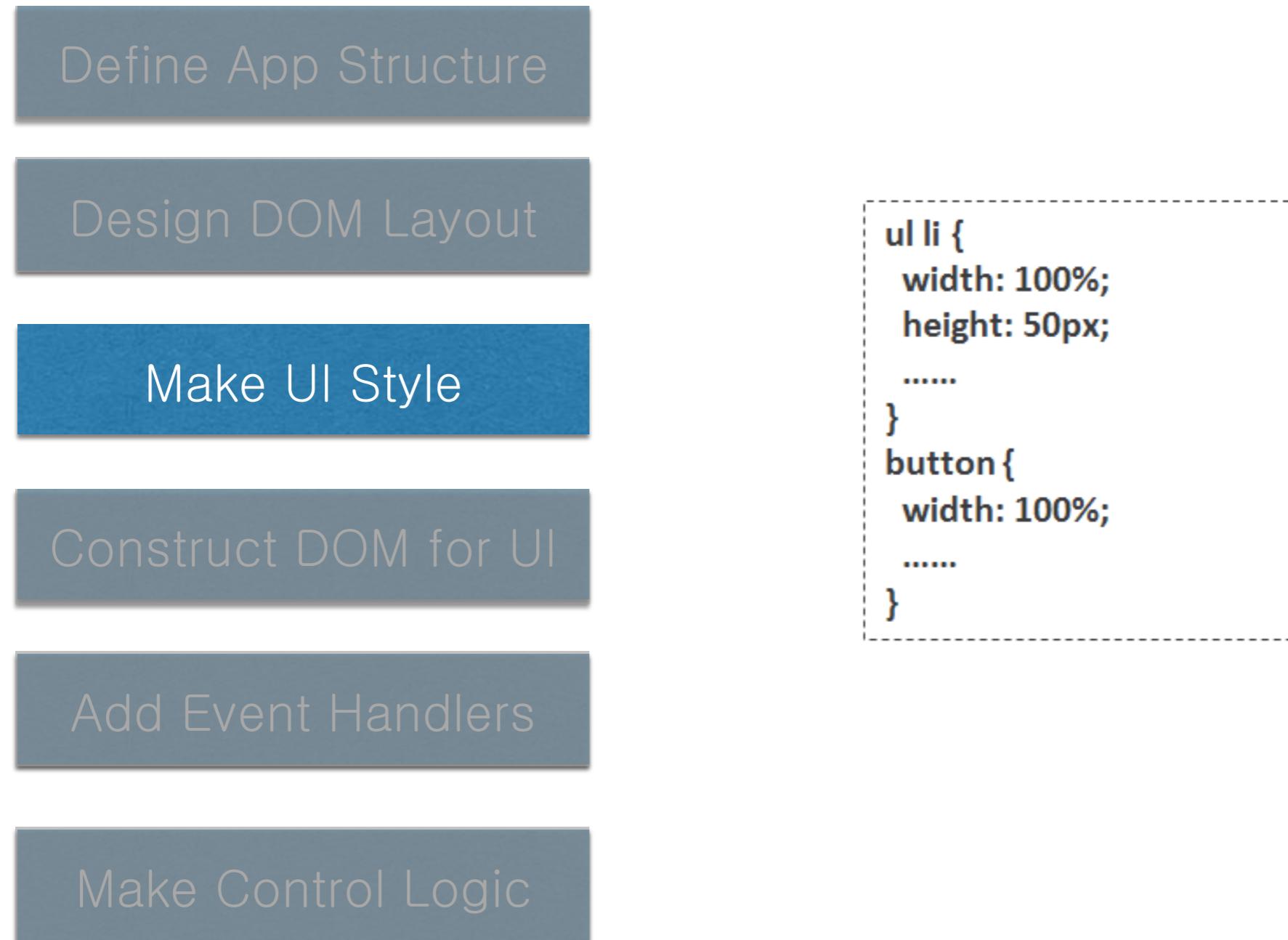
# TAU in Web Application Development Process

The following figure shows a typical workflow of Web application development.



# TAU in Web Application Development Process

The following figure shows a typical workflow of Web application development.



# TAU in Web Application Development Process

The following figure shows a typical workflow of Web application development.

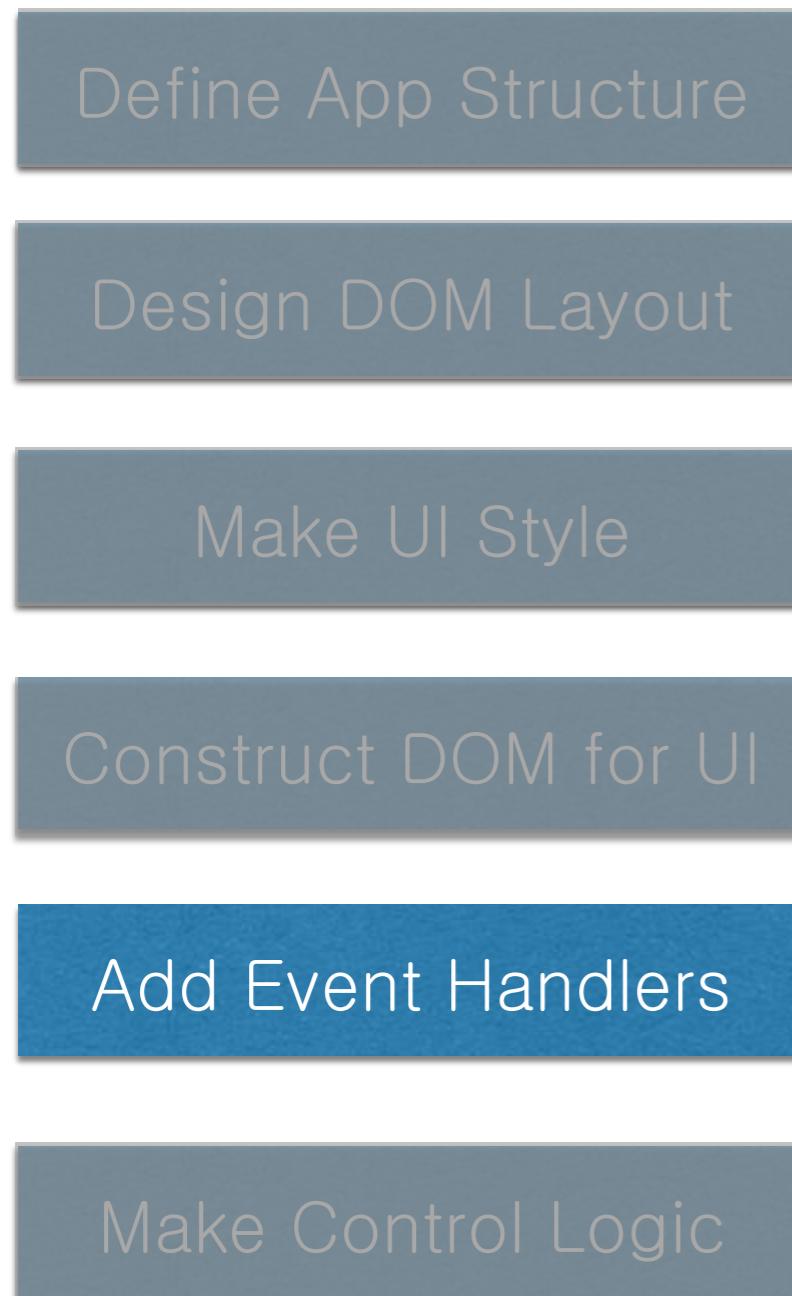


```
<div class="page">
<header>Phonebook List</header>
<div class="content">
  <ul id="list">
    <li>List1</li>
    <li>List2</li>
  </ul>
</div>
</div>
```



# TAU in Web Application Development Process

The following figure shows a typical workflow of Web application development.

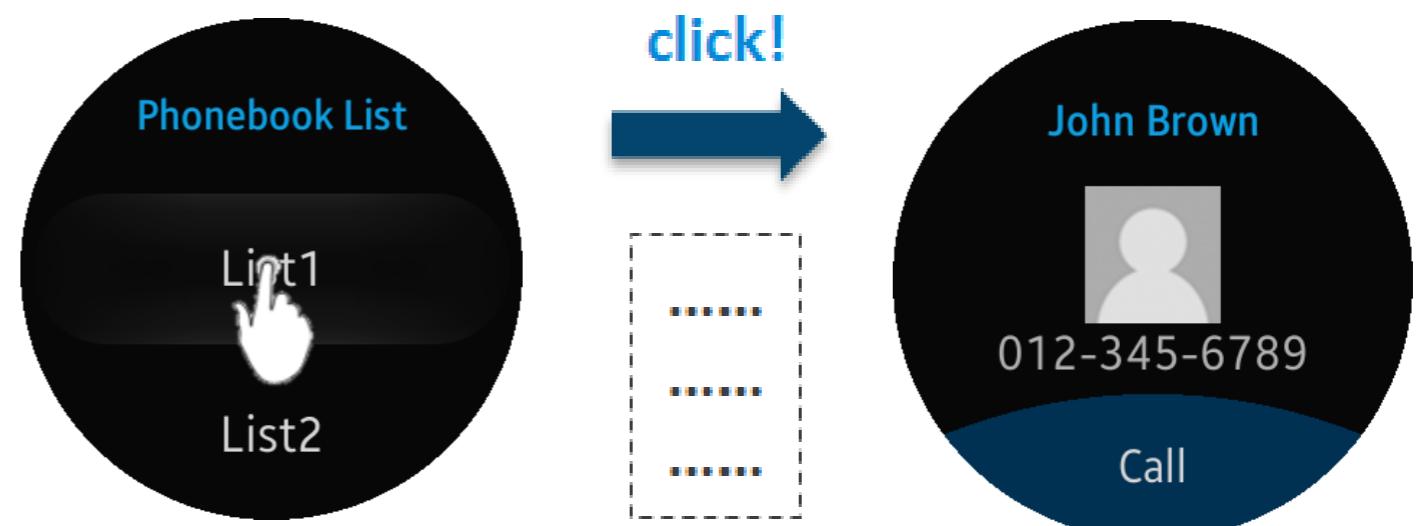
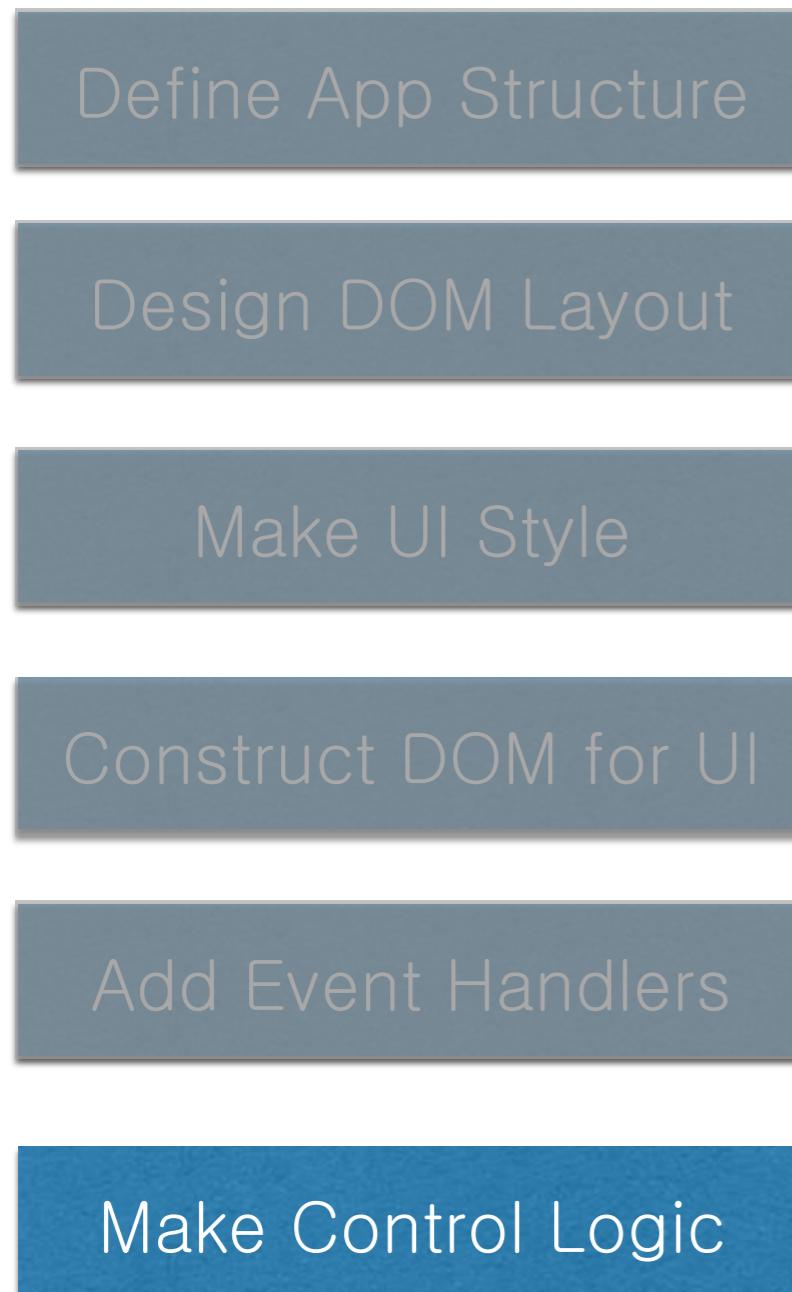


```
var list = document.getElementById("list");
list.addEventListener("click", function(e) {
    ....
});
```



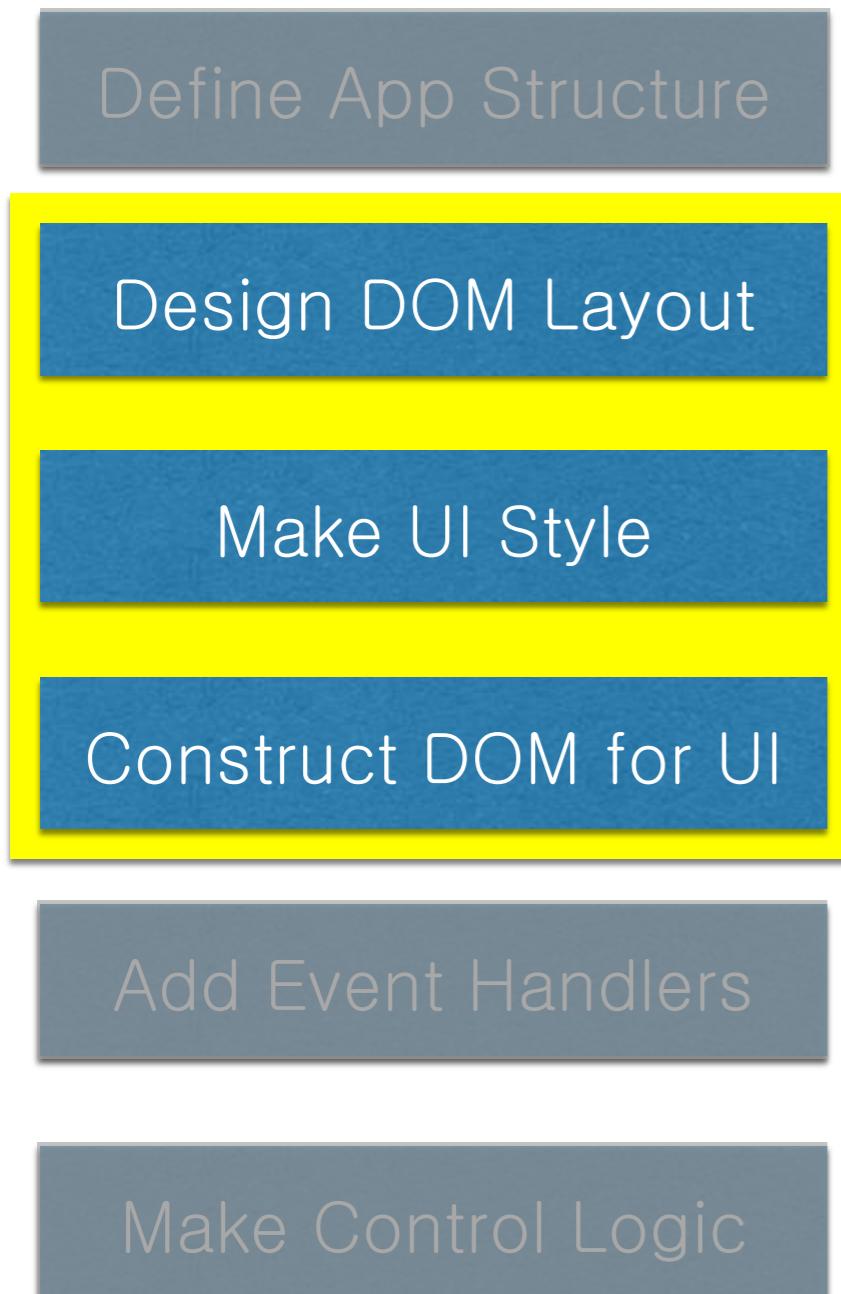
# TAU in Web Application Development Process

The following figure shows a typical workflow of Web application development.



# TAU in Web Application Development Process

The following figure shows a typical workflow of Web application development.



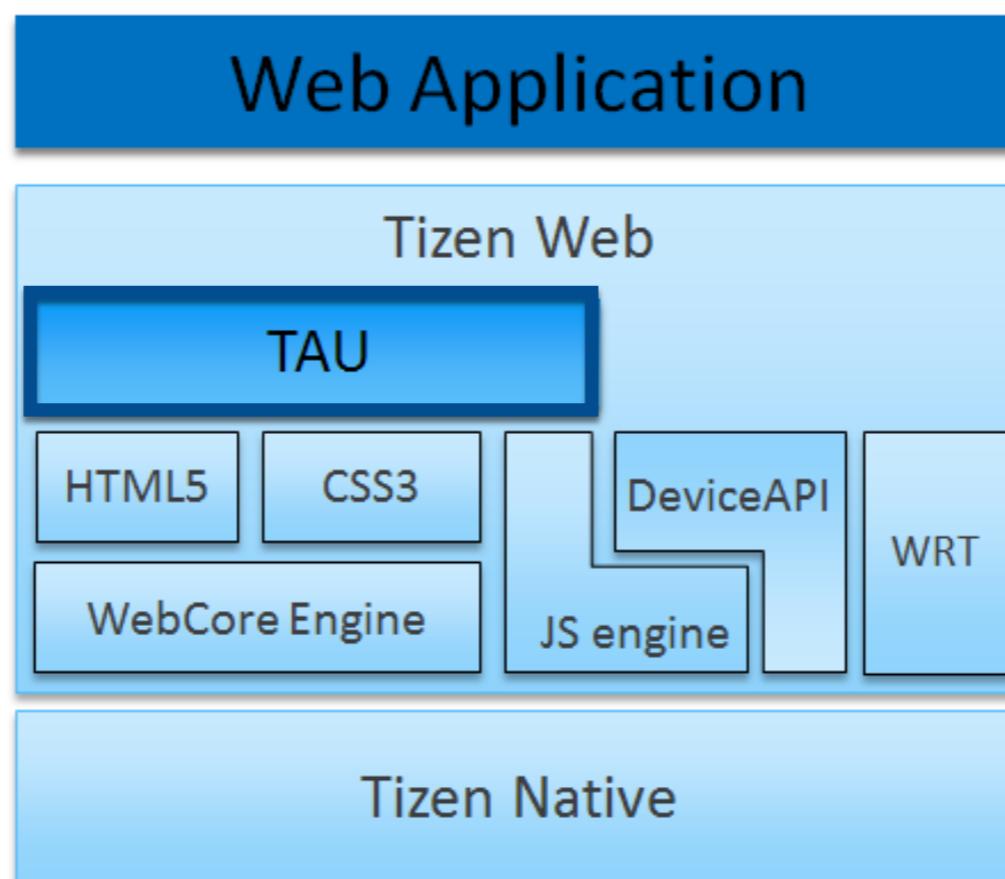
**TAU makes these steps easier.**

TAU supports the page layout, UI components, TAU APIs, and custom events.

# Core Concepts of TAU

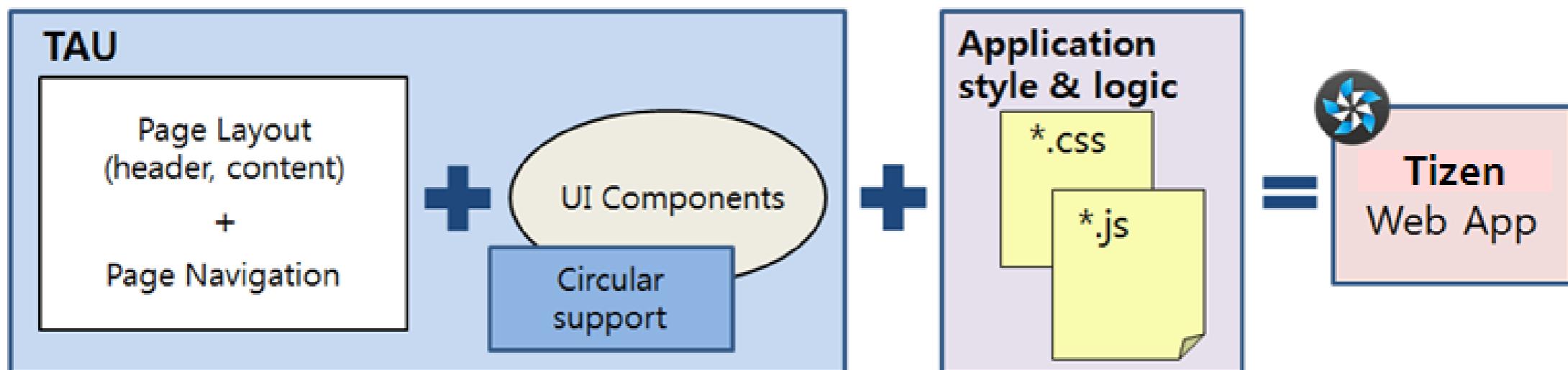
The concept of TAU is similar to jQuery Mobile, which is a commonly used Web UI system for making responsive Web sites and applications on mobile devices.

Compared to jQuery Mobile, TAU is much faster because it is made with pure JavaScript.



# Role of TAU

TAU supports the layout styles and page navigation of Web applications. TAU makes it easy to develop your own Tizen Web application with the UI components provided by TAU.

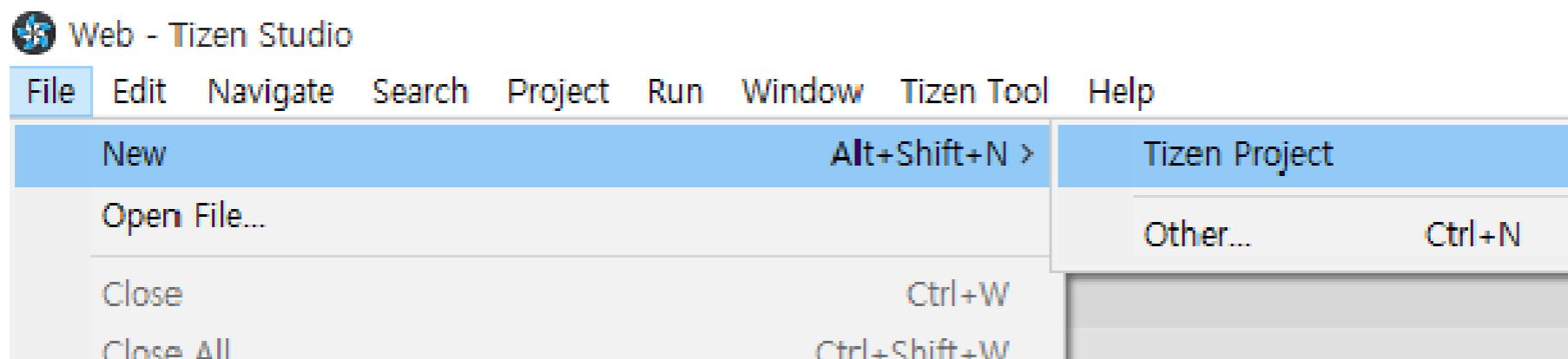


# Getting Started with TAU

# Getting Started with TAU

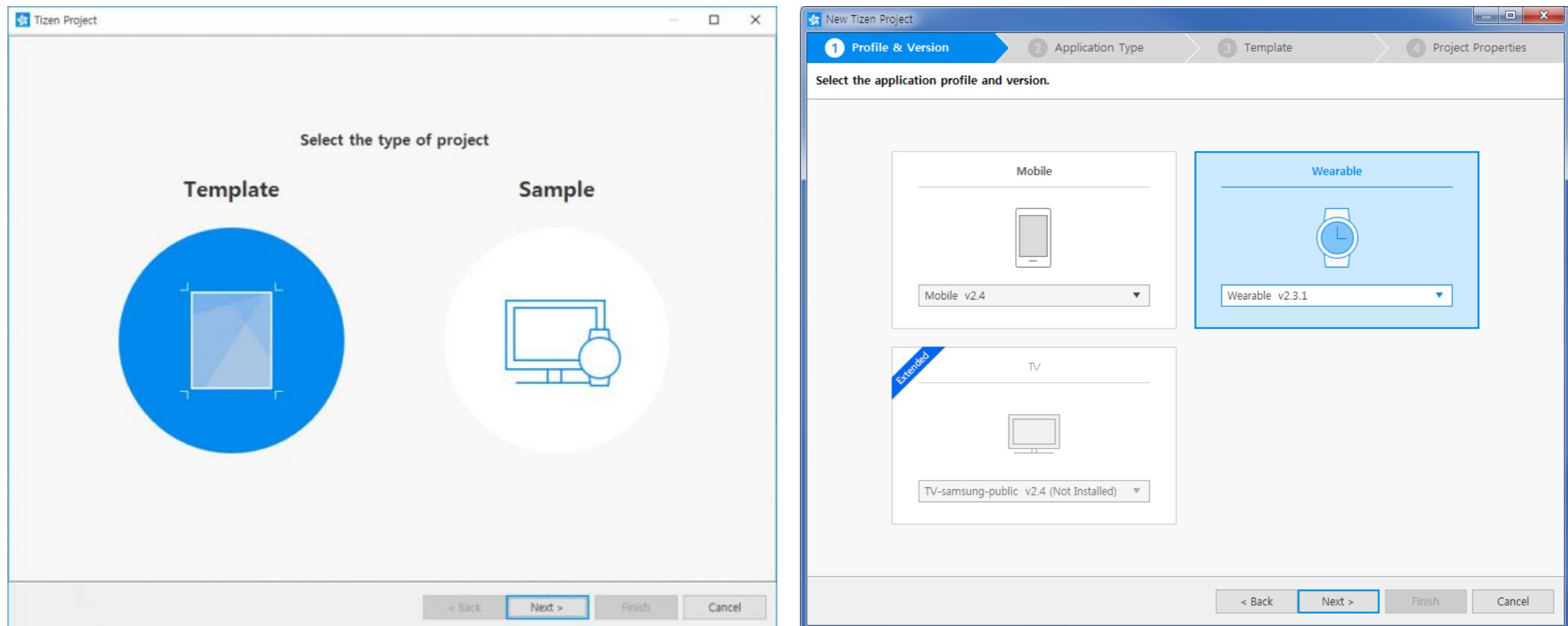
Create a new Web application project with the TAU library.

In the Tizen SDK, go to File > New > Tizen Web Project.



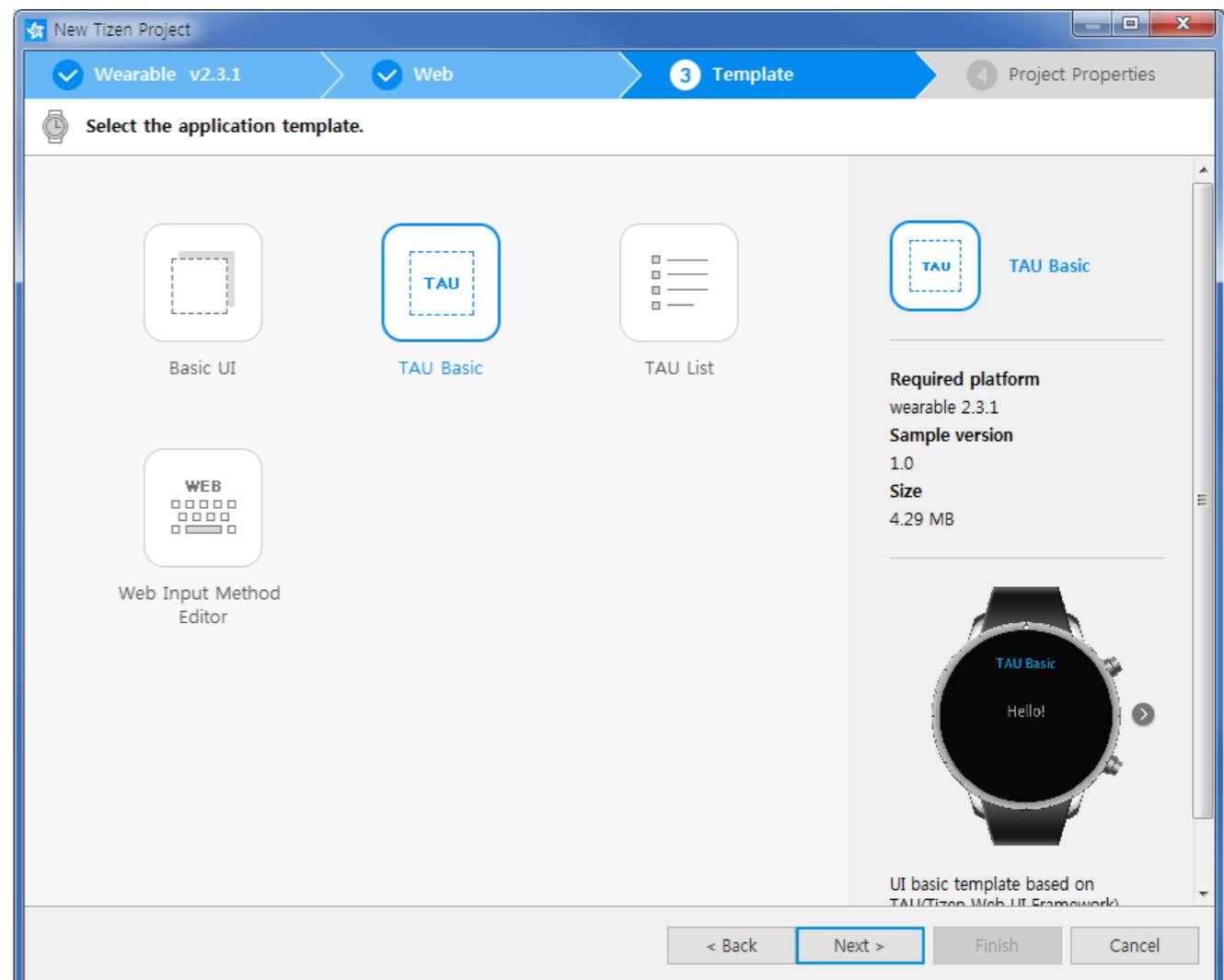
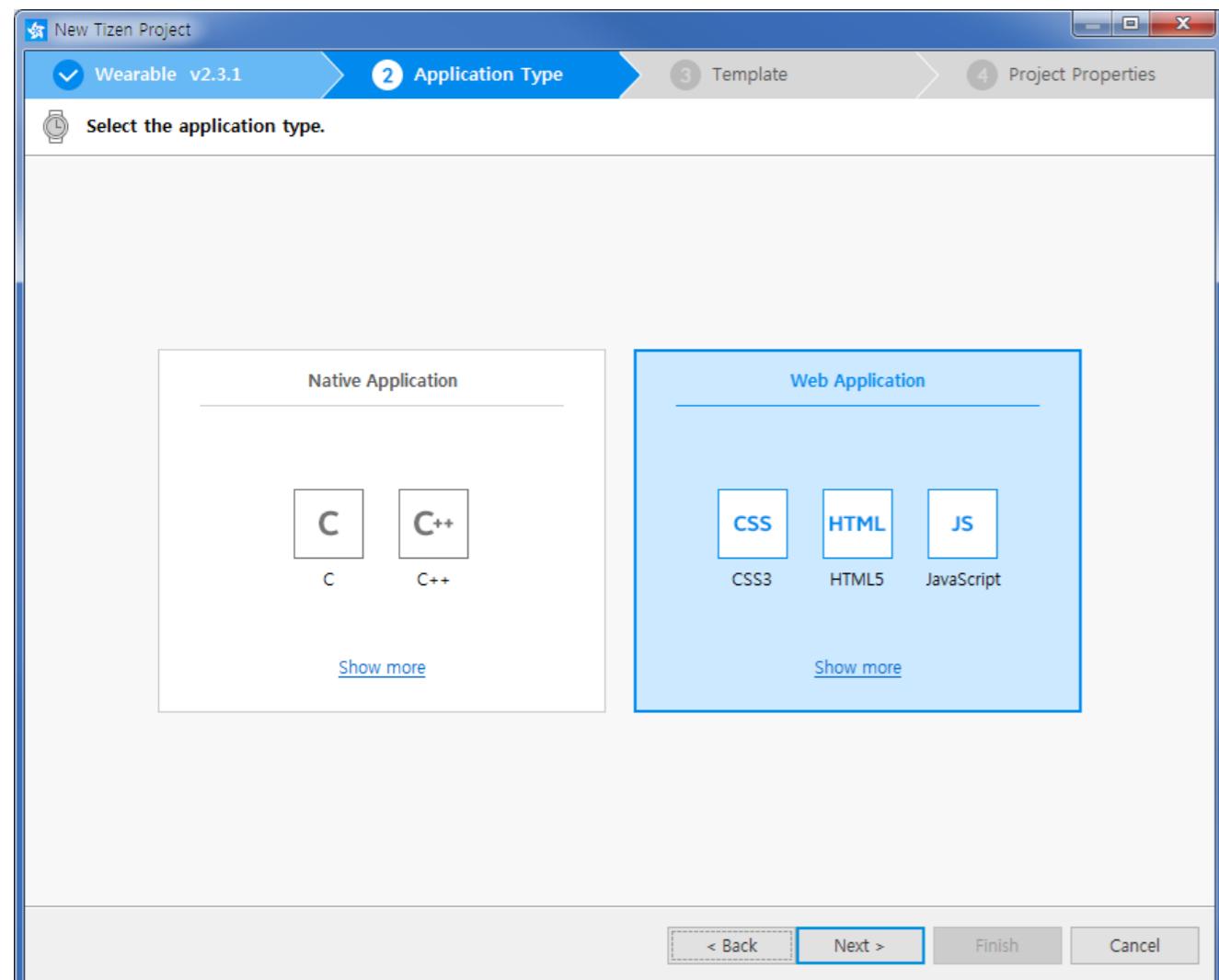
# Getting Started with TAU

In the project wizard pop-up window, select the **Template**, then select profile and versions: **WEARABLE v2.3.1** application.



# Getting Started with TAU

Select Web Application and click Next. Then select the application template.

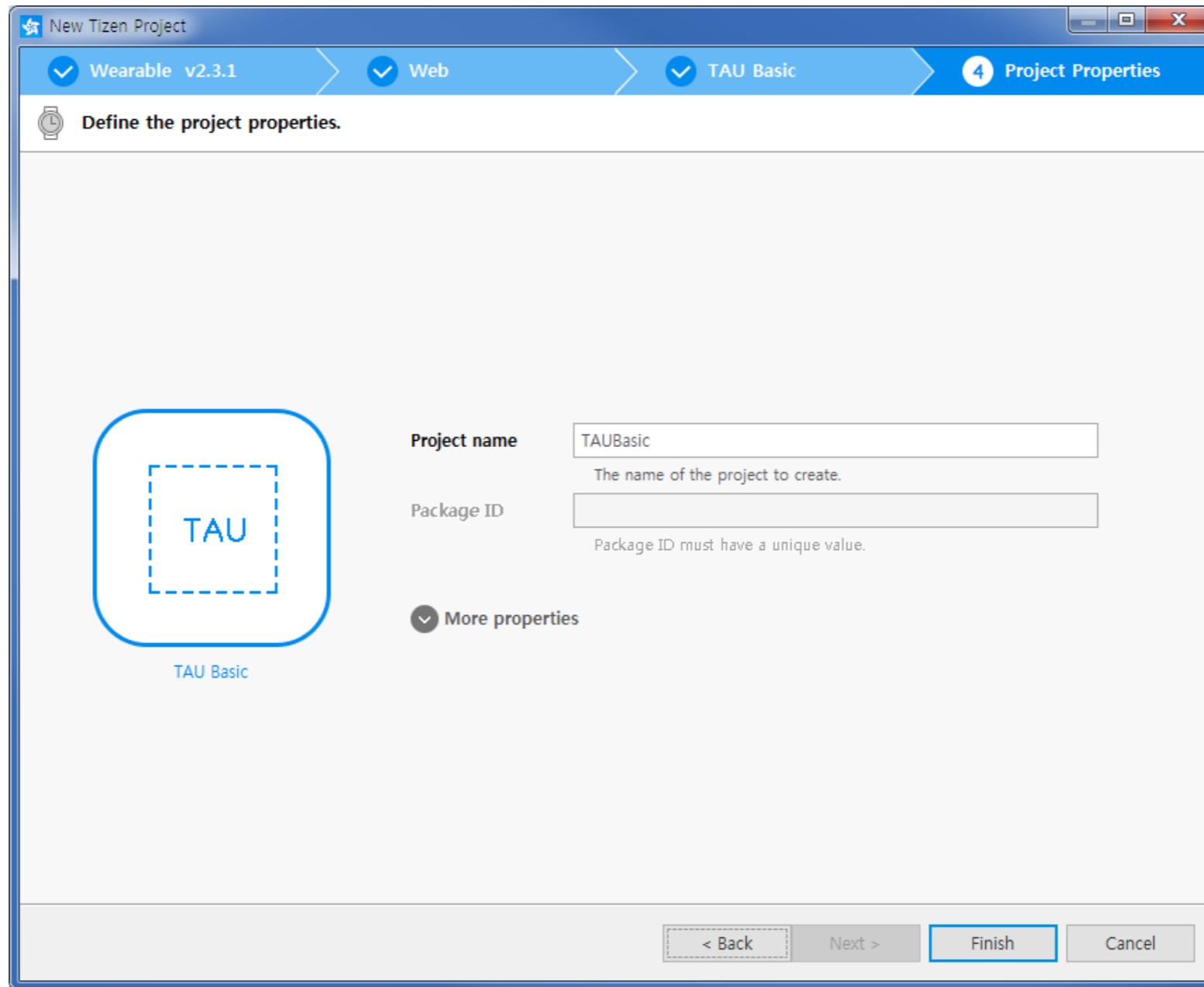


# Getting Started with TAU

Change the project name, if you want.

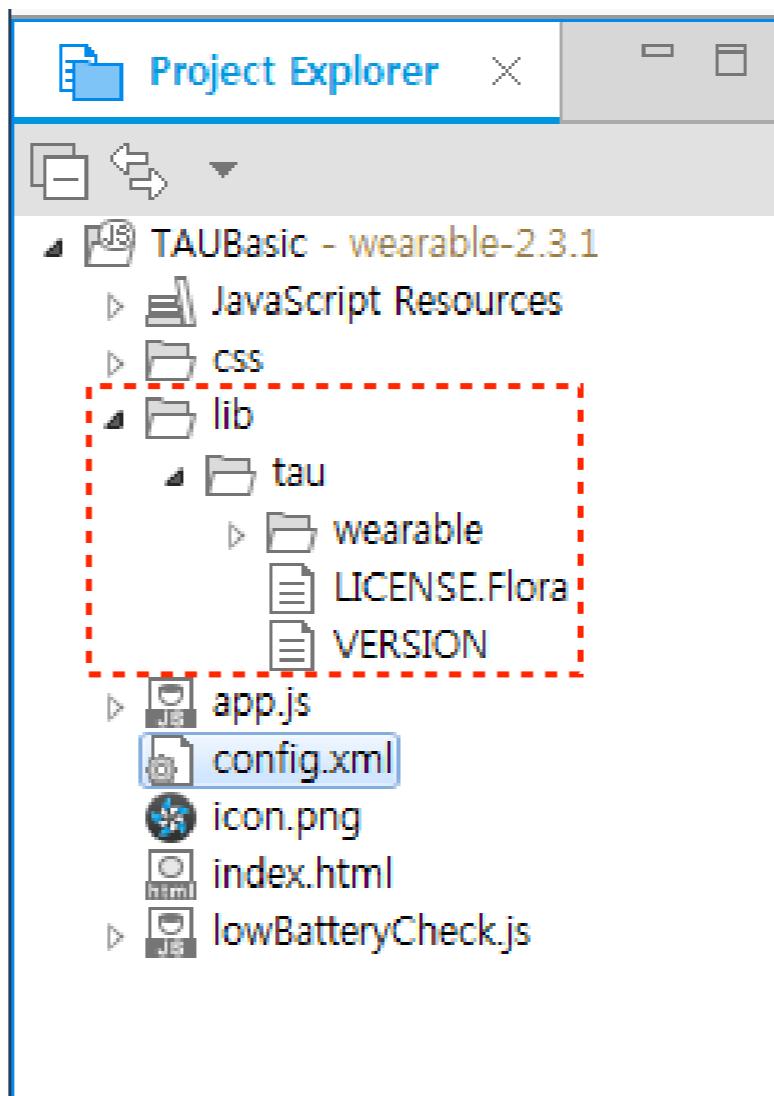
Click More properties, if you want to change more properties.

Leave all other fields in the Project Wizard to their default values, and click Finish.



# Getting Started with TAU

The TAU library is included in the project folders.



# Getting Started with TAU

The TAU library consists of the following CSS and Javascript files.

File	Description
tau(.min).js	Mandatory The main file required to use the TAU Interface
tau(.min).css	Mandatory The style sheet of the TAU theme
tau.circle(.min).css	Optional Required for developing a Web application with a circular UI
circle-helper.js	Optional Required for developing a Web application with a circular UI

TAU library

App JS provided by TAU

# Getting Started with TAU

In the index.html file, you can see how to connect TAU.

```
<!DOCTYPE html>
<html>
<head>
    <meta name="viewport" content="width=device-width, user-scalable=no">
    <title>Circular UI</title>
    <link rel="stylesheet" href="lib/tau/wearable/theme/default/tau.css"/>
    <!-- For Circular Device please import tau.circle.css -->
    <link rel="stylesheet" media="all and (-tizen-geometric-shape: circle)"
          href="lib/tau/wearable/theme/default/tau.circle.css">
</head>
<body>
    <!-- HTML BODY CONTENT -->
    <script type="text/javascript" src="lib/tau/wearable/js/tau.js"></script>
    <script type="text/javascript" src="js/circle-helper.js"></script>
</body>
</html>
```

Connect tau.js  
at the end

Connect  
tau.css

For developing applications with circular UI,  
connect tau.circle.css and circle-helper.js

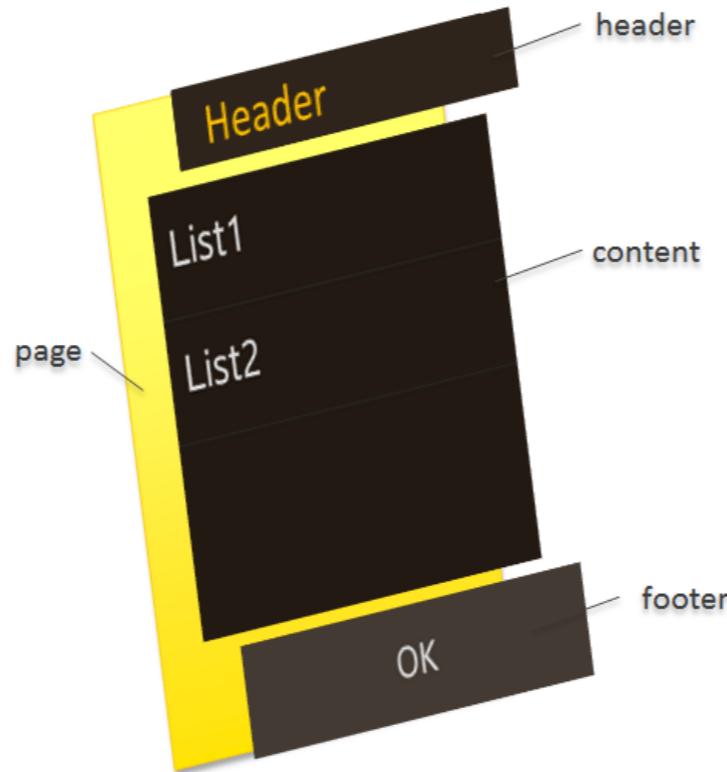
For better performance,

- CSS files must be included in the header.
- JavaScript files must be put before the closing tag of the body element.

# Creating Simple UI with TAU

# Creating Simple UI with TAU – Page Structure

The following figure illustrates the page structure using TAU.



A TAU page consists of:

- **Header** section (optional) displaying the title of the page,
- **Content** section displaying the main content of the page,
- **Footer** section (optional) displaying the bottom buttons.

# Creating Simple UI with TAU – Page

Create a simple Page using TAU.

A Page component can be created by adding a `<div>` element with `.ui-page` class in the `index.html` file.

After creating a Page component, add a `<header>` element with `.ui-header` class and a `<div>` content element with `.ui-content` class inside the Page component.

`index.html`

```
<body>
  <div class="ui-page" id="main">
    <header class="ui-header">
      <h2 class="ui-title">Hello World</h2>
    </header>
    <div class="ui-content">
      <p>This is a content area</p>
    </div>
  </div>
  ...
</body>
```

Add a TAU page  
with header and content



# Creating Simple UI with TAU – Listview

Create a simple Listview component using TAU.

A Listview component can be created by adding a `<ul>` element with the `.ui-listview` class inside the content element.

Each item of the list can be created by adding a `<li>` element inside the Listview element, and by using the `<a>` tag, it can be linked to another page.

index.html

```
<body>
  <div class="ui-page" id="main">
    <header class="ui-header">
      <h2 class="ui-title">Hello World</h2>
    </header>
    <div class="ui-content">
      <ul class="ui-listview">
        <li><a href="#">List Item1</a></li>
        <li><a href="#">List Item2</a></li>
        <li><a href="#">List Item3</a></li>
        <li><a href="#">List Item4</a></li>
      </ul>
    </div>
  ...
</body>
```



Add a simple TAU listview

# Creating Simple UI with TAU – SnapListview

The SnapListview component is specialized for the circular UI. With it, the list can be scrolled using the bezel on the edge of a circular wearable devices.





# Creating Simple UI with TAU – SnapListview

To create an advanced SnapListview component which is specialized for circular UI, some additional codes are required.

TAU provides the codes through the "circle-helper.js" file.

js/circle-helper.js

```
(function(tau) {  
    var page,  
        list,  
        listHelper;  
  
    if (tau.support.shape.circle) {  
        document.addEventListener("pagebeforeshow", function (e) {  
            page = e.target;  
            list = page.querySelector(".ui-listview");  
        // Create SnapListView and binding rotary event using tau.helper  
        if (list) {  
            listHelper = tau.helper.SnapListStyle.create(list);  
        }  
    });  
});
```

Add new variables

Grabs a HTML element with the class name 'ui-listview' and stores the element object in the variable 'list'

Create SnapListView and bind rotary event using tau.helper

continue on the next page..

# Creating Simple UI with TAU – SnapListView

js/circle-helper.js

```
document.addEventListener("pagebeforehide", function (e) {  
    if(list) {  
        listHelper.destroy();  
    }  
});  
}(tau));
```

Destroy the SnapListView and unbind rotary event when the list page is hidden



# Creating Simple UI with TAU – Page Navigation

Create another page and link the new page with the List item1.

index.html

```
<body>
  <div class="ui-page" id="main">
    ...
    <div class="ui-content">
      <ul class="ui-listview">
        <li><a href="#item1">List Item1</a></li>
        <li><a href="#">List Item2</a></li>
        <li><a href="#">List Item3</a></li>
        <li><a href="#">List Item4</a></li>
      </ul>
    </div>
  </div>
  <div class="ui-page" id="item1">
    <header class="ui-header">
      <h2 class="ui-title">Item1</h2>
    </header>
    <div class="ui-content">
      <p>Content for Item1 page</p>
    </div>
  </div>
  ...
</body>
```

Link the first item of the list to the page with the id 'item1'

create a new page with the id 'item1'

# Creating Simple UI with TAU – Page Navigation

- (1) When the first item of the list is clicked, the view moves to the Item1 page.
- (2) When the back button is clicked, the view returns to the SnapListview page.



TAU provides transition effect for pages. For more information, see:

[https://developer.tizen.org/dev-guide/2.3.1/org.tizen.web.apireference/html/ui\\_fw\\_api/wearable/page/page\\_change.htm](https://developer.tizen.org/dev-guide/2.3.1/org.tizen.web.apireference/html/ui_fw_api/wearable/page/page_change.htm)

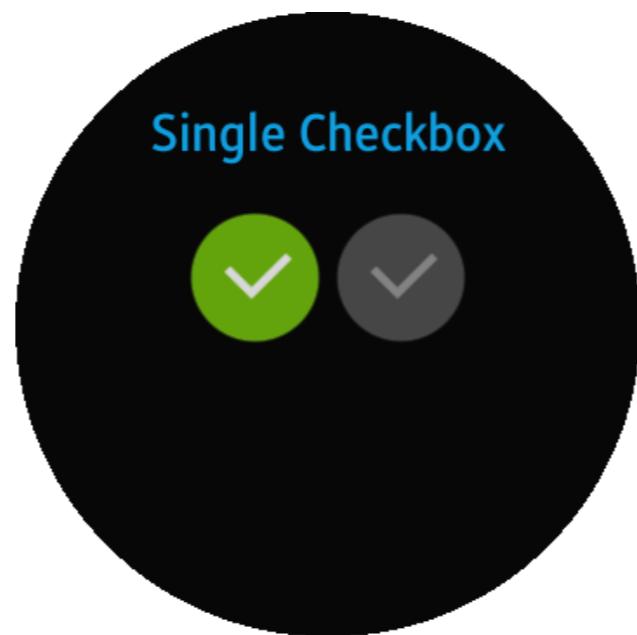
# Creating Simple UI with TAU – Checkbox

To create a checkbox component, you don't have to use specific classes.

index.html

```
<body>
  <div class="ui-page" id="main">
    <div class="ui-header">
      <h2 class="ui-title">Single Checkbox</h2>
    </div>
    <div class="ui-content">
      <input type="checkbox" checked/>
      <input type="checkbox">
    </div>
  </div>
  ...
</body>
```

use input element with a  
checkbox type



# Creating Simple UI with TAU – Progress

To create a progress component, a `<progress>` element with the "ui-circle-progress" class and JavaScript code are required.

`index.html`

```

<body>
  <div class="ui-page" id="main">
    <div class="ui-content">
      Progress Sample
      <progress class="ui-circle-progress" id="circleprogress" max="100" value="20"></progress>
    </div>
  </div>
  ...
</body>

```

→ Declare a progress tag with the "ui-circle-progress" class.

`js code`

```

(function(tau) {
var page = document.getElementById("pageCircleProgressBar"),
elProgressBar = document.getElementById("circleprogress"),
progressBar;

pageBeforeShowHandler = function() {
  progressBar = tau.widget.CircleProgressBar(elProgressBar, {size: "full"});
  progressBar.value(elProgressBar.value);
};

page.addEventListener("pagebeforeshow", pageBeforeShowHandler);

/* You can use additional event listeners here */
}(tau));

```

create a tau circle progress bar component and set the value.

This is an example of full size progress component.

# Creating Simple UI with TAU – Progress

js code (the same as the previous page)

```
(function(tau) {
var page = document.getElementById("pageCircleProgressBar"),
elProgressBar = document.getElementById("circleprogress"),
progressBar;

pageBeforeShowHandler = function() {
    progressBar = tau.widget.CircleProgressBar(elProgressBar, {size: "full"});
    progressBar.value(elProgressBar.value);
};

page.addEventListener("pagebeforeshow", pageBeforeShowHandler);

/* You can use additional event listeners here */
})(tau));
```

TO control the progress value with some interactions, you can use additional event listeners,

such as button click or rotary event.



# Appendix

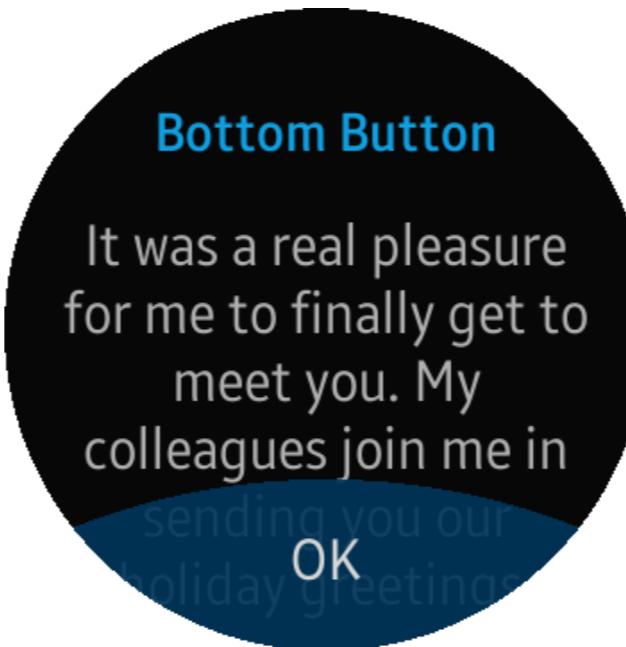
- TAU UI Components
- References

# TAU UI Components (1)

TAU provides a variety of UI Components.



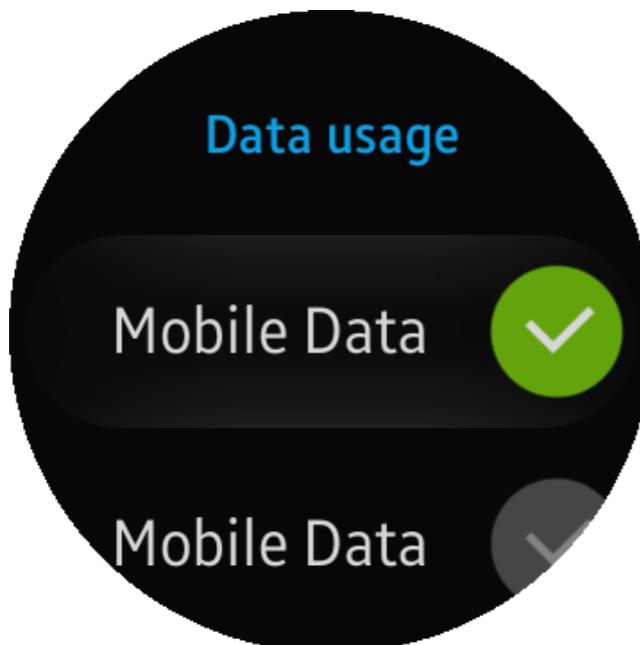
Header



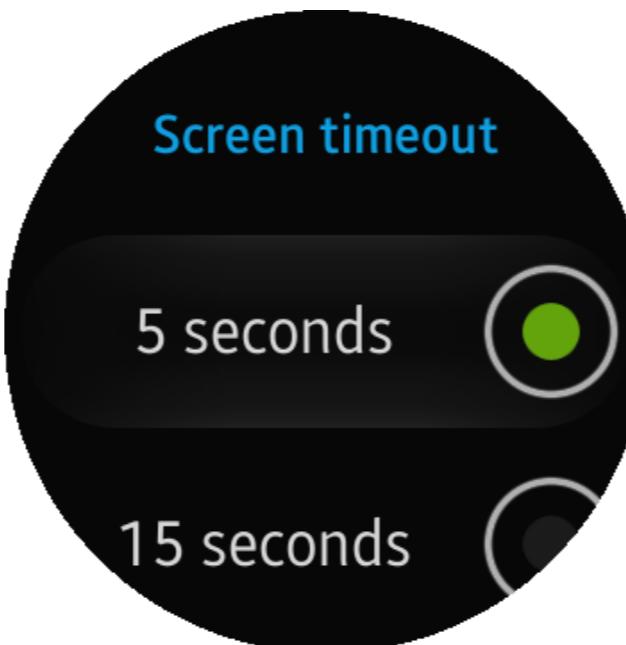
Bottom Button



Listview / SnapListview



Checkbox



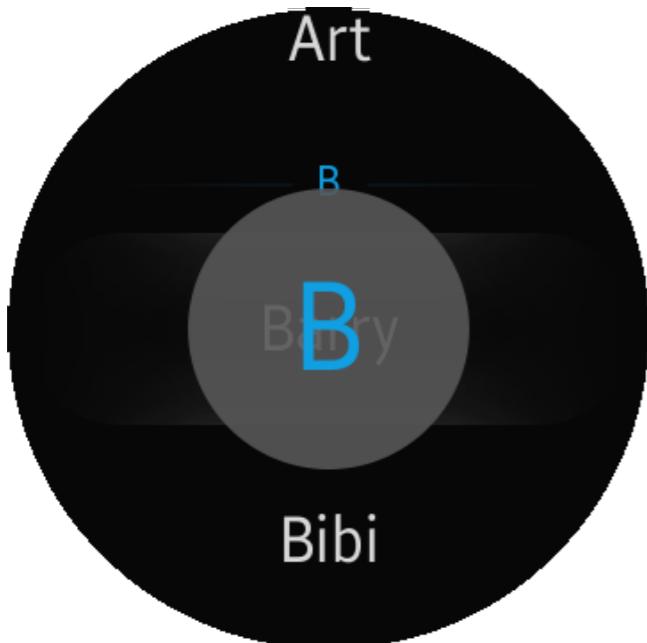
Radio



Toggle

# TAU UI Components (2)

TAU provides a variety of UI Components.



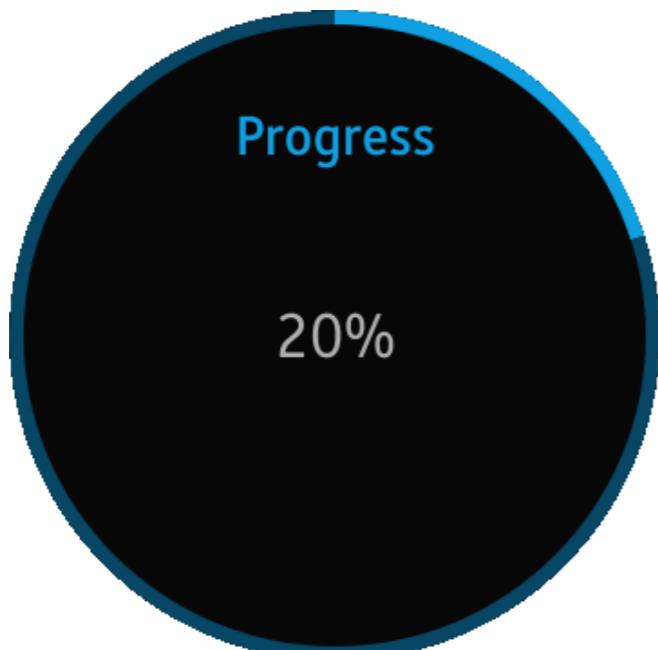
Index Scrollbar



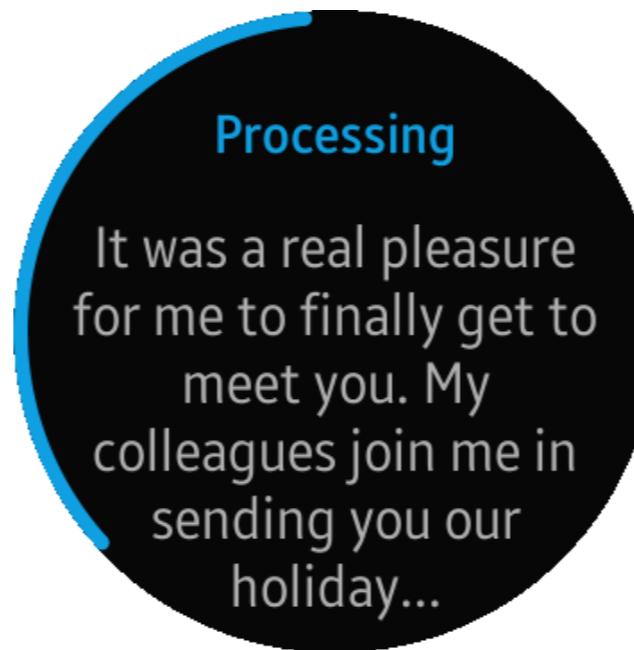
Indicator



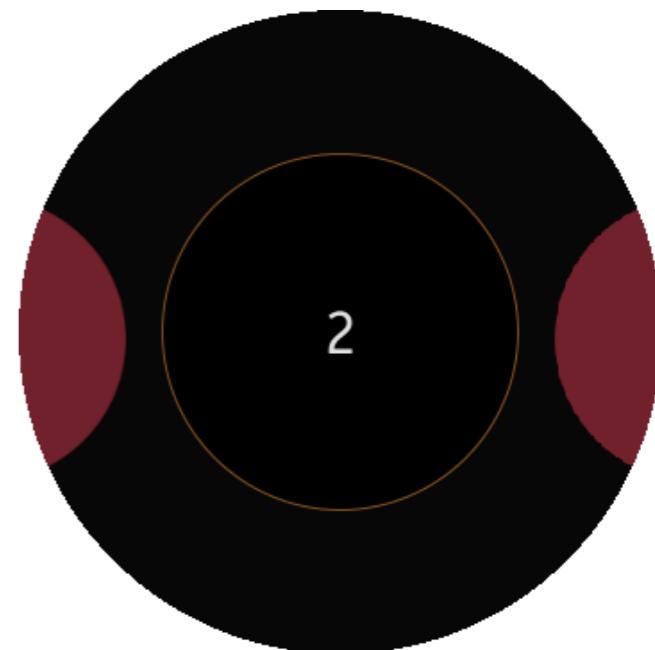
Marquee



Progress



Processing



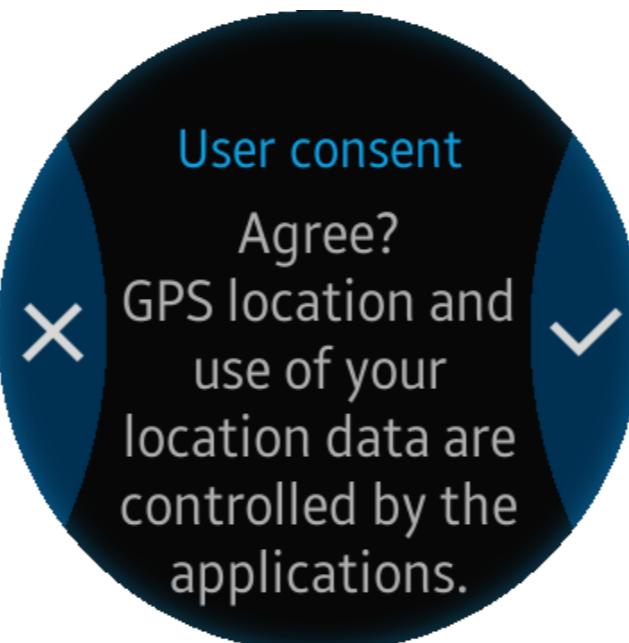
Section Changer  
(Thumbnail)

# TAU UI Components (3)

TAU provides a variety of UI Components.



1-button Popup



2-button Popup



Toast Popup



Graphic  
Toast Popup

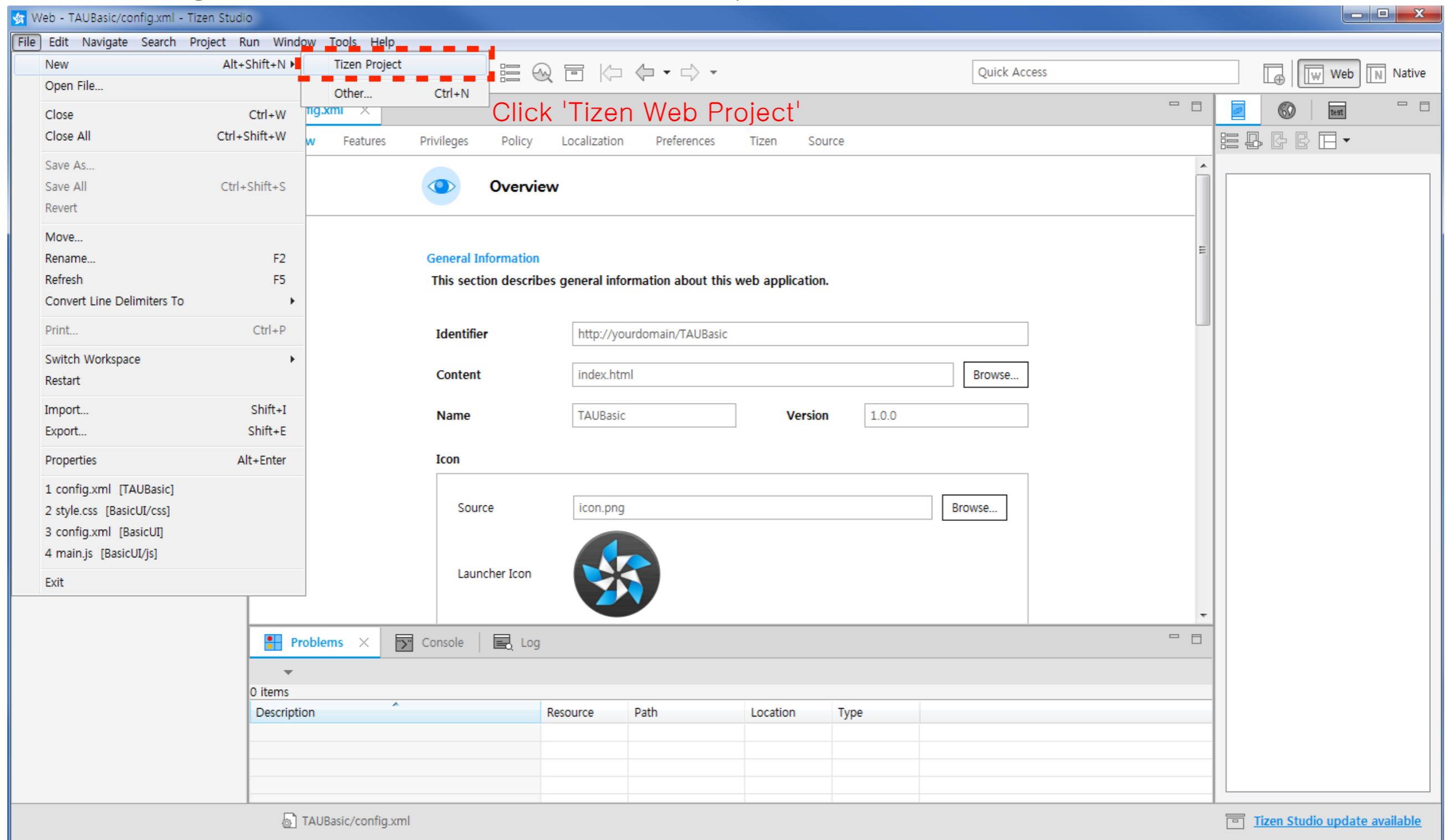


Selector

# TAU UI Components Sample Application

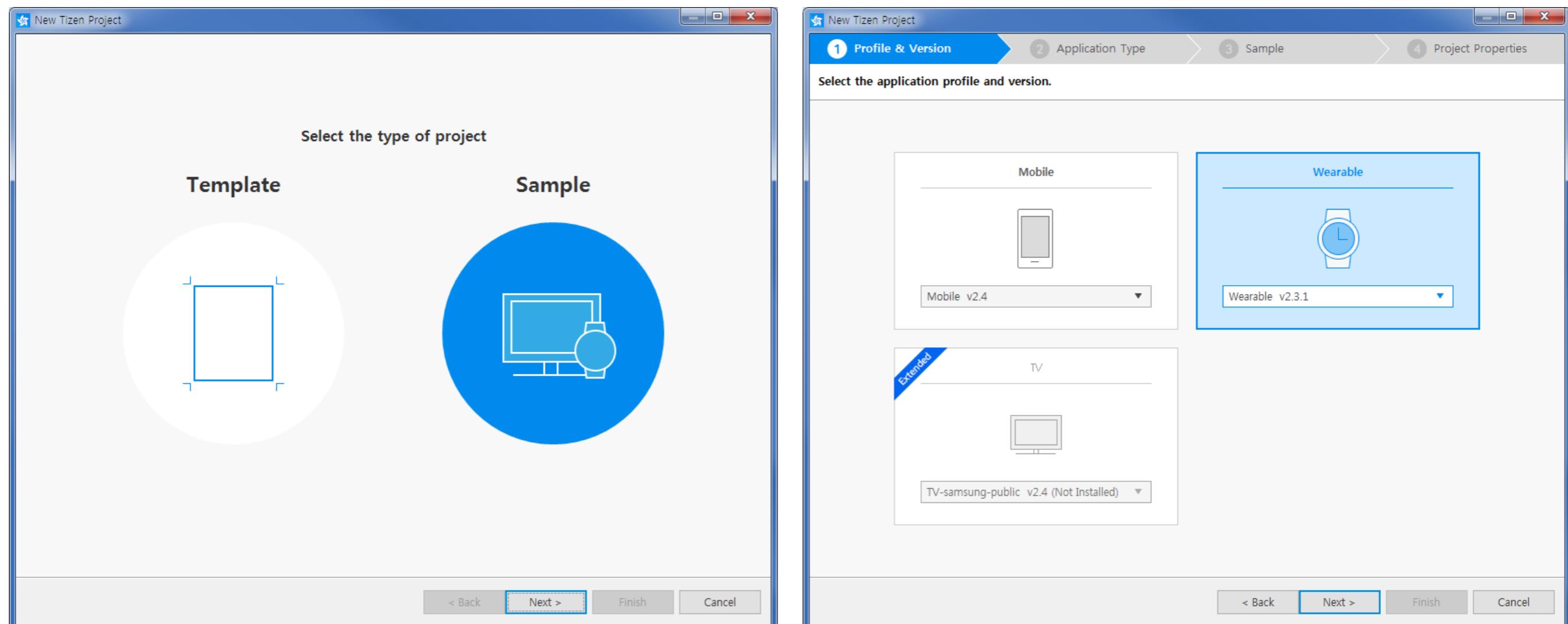
TAU provides UI components sample applications for wearable and mobile.

Following shows how to create the sample.



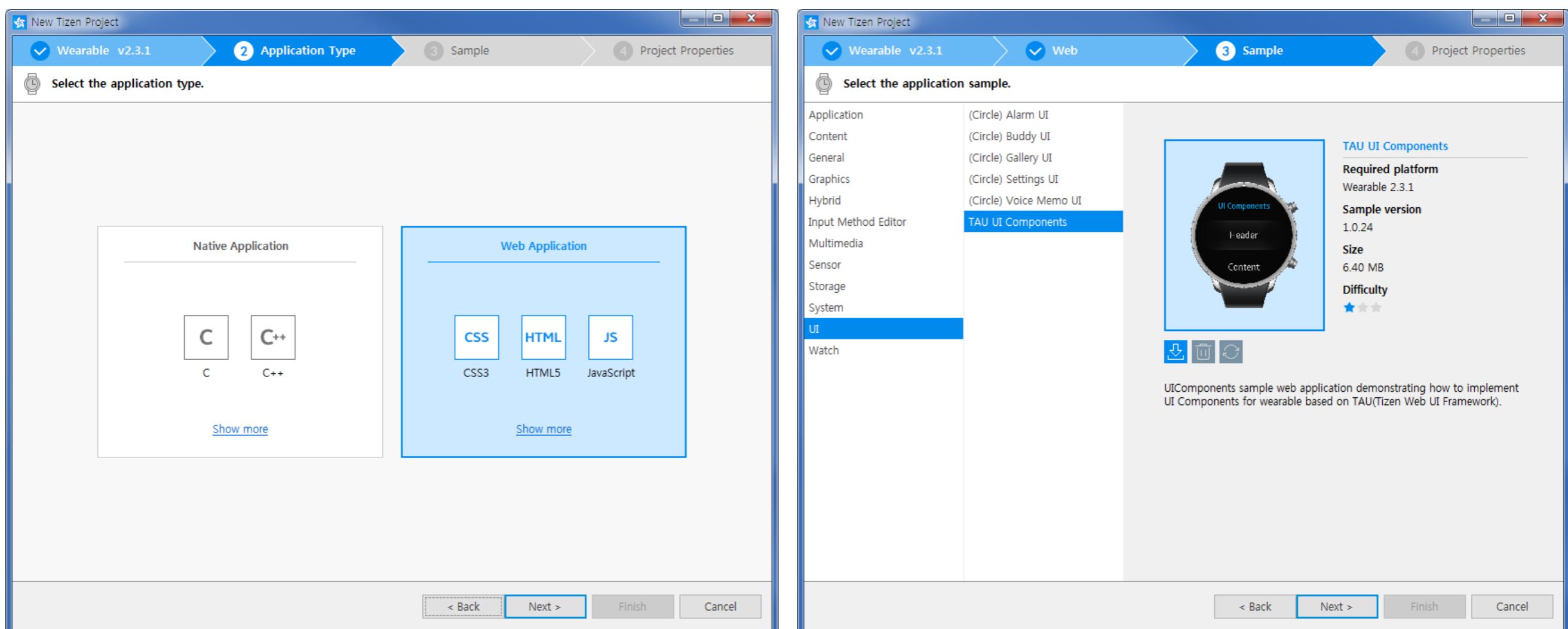
# TAU UI Components Sample Application

TAU provides UI components sample applications for wearable and mobile.  
This is a wearable sample.



# TAU UI Components Sample Application

TAU provides UI components sample applications for wearable and mobile.  
This is a wearable sample.



# TAU references

For more information about TAU and using the UI Components, see the following:

- TAU API Reference Guide(Wearable)

[https://developer.tizen.org/dev-guide/2.3.1/org.tizen.web.apireference/html/ui\\_fw\\_api/wearable/index.htm](https://developer.tizen.org/dev-guide/2.3.1/org.tizen.web.apireference/html/ui_fw_api/wearable/index.htm)

- TAU API Reference Guide(Mobile)

[https://developer.tizen.org/dev-guide/2.4/org.tizen.web.apireference/html/ui\\_fw\\_api/ui\\_fw\\_api\\_cover.htm](https://developer.tizen.org/dev-guide/2.4/org.tizen.web.apireference/html/ui_fw_api/ui_fw_api_cover.htm)

2.4 > API References > Web

Application > Tizen Advanced UI

- TAU UI Practice

<https://developer.tizen.org/development/ui-practices/web-application/tau/>

Web Application > TAU

