



Easily Port Web Apps to Tizen & Develop with Tizen Device APIs

Qingqiu Li
Samsung

Content

1. Tizen Web Application
2. Porting an HTML5 App to Tizen IDE
3. Tizen Web Device APIs
4. TAU Introduction

An abstract graphic on the left side of the slide. It features several triangles of varying sizes, some with blue and white horizontal stripes and others solid white. A trail of small, colorful dots (blue, green, and dark blue) extends from the bottom of the striped triangles towards the bottom right corner of the slide.

Tizen Web Application

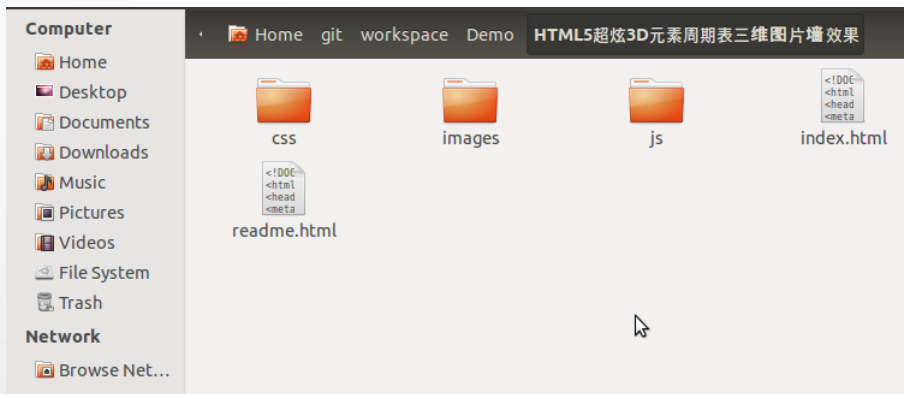
Overview on web application



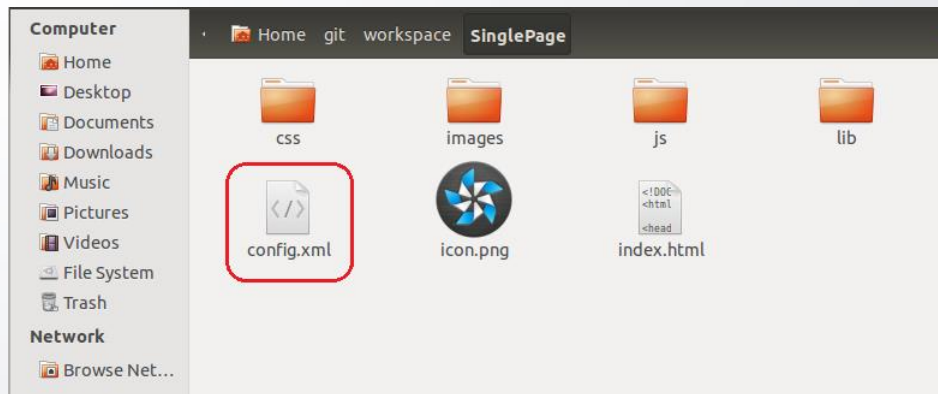
HTML5 App & Tizen Web App (1/3)

Difference in file structure :

HTML5 App



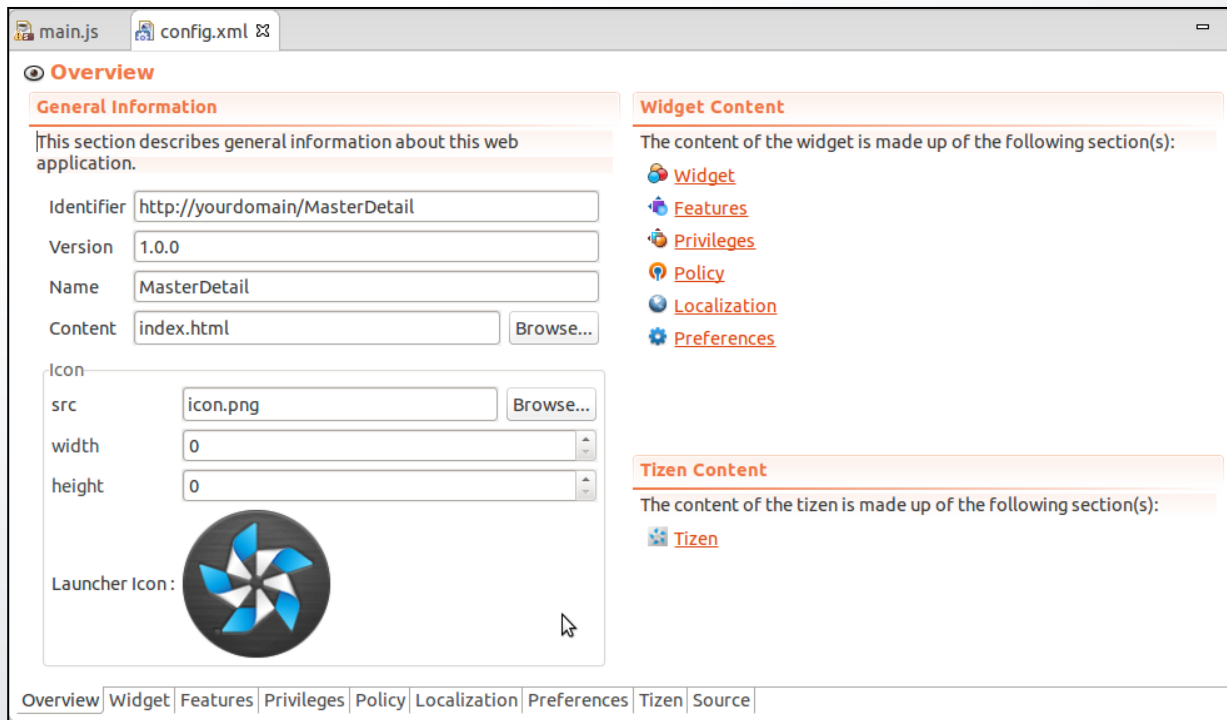
Tizen Web App (with configure file)



HTML5 App & Tizen Web App (2/3)

Configure File :

User Interface, modify the features, privileges, policy, and etc.



The screenshot displays the Tizen Web App configuration tool interface. At the top, there are tabs for 'main.js' and 'config.xml'. The 'Overview' tab is selected, showing a summary of the application configuration. The 'General Information' section includes fields for Identifier, Version, Name, and Content. The 'Widget Content' section lists the components included in the widget. The 'Tizen Content' section lists the components included in the Tizen application. The 'Launcher Icon' section shows a preview of the application icon.

Overview

General Information

This section describes general information about this web application.

Identifier:

Version:

Name:


Content: [Browse...](#)

Icon

src: [Browse...](#)

width:

height:

Launcher Icon: 

Widget Content

The content of the widget is made up of the following section(s):

- [Widget](#)
- [Features](#)
- [Privileges](#)
- [Policy](#)
- [Localization](#)
- [Preferences](#)

Tizen Content

The content of the tizen is made up of the following section(s):

- [Tizen](#)

[Overview](#) [Widget](#) [Features](#) [Privileges](#) [Policy](#) [Localization](#) [Preferences](#) [Tizen](#) [Source](#)

HTML5 App & Tizen Web App (3/3)

Configure File :

Source Code, modify the features, privileges, policy, and etc.

```
<access origin="*" subdomains="true"/>
```

```
<?xml version="1.0" encoding="UTF-8" ?>
<widget xmlns="http://www.w3.org/ns/widgets" xmlns:html="http://www.w3.org/1999/xhtml"
  <access origin="*" subdomains="true"/>
  <tizen:application id="2ixfkqTgBW.btconn" package="2ixfkqTgBW" required_version="2.3"/>
  <content src="index.html"/>
  <feature name="http://tizen.org/feature/screen.size.all"/>
  <feature name="http://tizen.org/feature/camera"/>
  <name>btconn</name>
  <tizen:privilege name="http://tizen.org/privilege/application.launch"/>
  <tizen:privilege name="http://tizen.org/privilege/bluetooth.admin"/>
  <tizen:privilege name="http://tizen.org/privilege/bluetooth.gap"/>
  <tizen:privilege name="http://tizen.org/privilege/bluetooth.spp"/>
  <tizen:profile name="wearable"/>
  <tizen:setting background-support="disable" encryption="disable" hwkey-event="enable"/>
</widget>
```

```
<tizen:privilege name="http://tizen.org/privilege/application.launch"/>
<tizen:privilege name="http://tizen.org/privilege/bluetooth.admin"/>
<tizen:privilege name="http://tizen.org/privilege/bluetooth.gap"/>
<tizen:privilege name="http://tizen.org/privilege/bluetooth.spp"/>
```

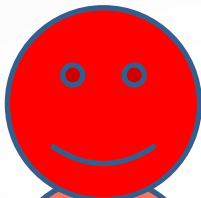
An abstract graphic on the left side of the slide. It features several triangles of varying sizes, some with blue and white horizontal stripes and others solid white. These triangles are arranged in a way that suggests movement or a cluster. Scattered around and below the triangles are numerous small, colorful dots in shades of blue, green, and purple. The background of the entire slide is a light gray with subtle, concentric circular patterns.

Porting an HTML5 App to Tizen IDE

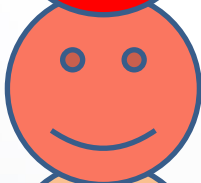
Common Porting Step :

1. Create a Web Project in SDK
2. Copy Web App to SDK Project
3. Modify configure file
4. Run project in Tizen Target

Why to port HTML5 Apps to Tizen IDE?



Tizen Platform has more chance than other platform, and porting is so easy.



Tizen Platform support HTML5 technique very well.



HTML5 technique is common, can be used in different platform.

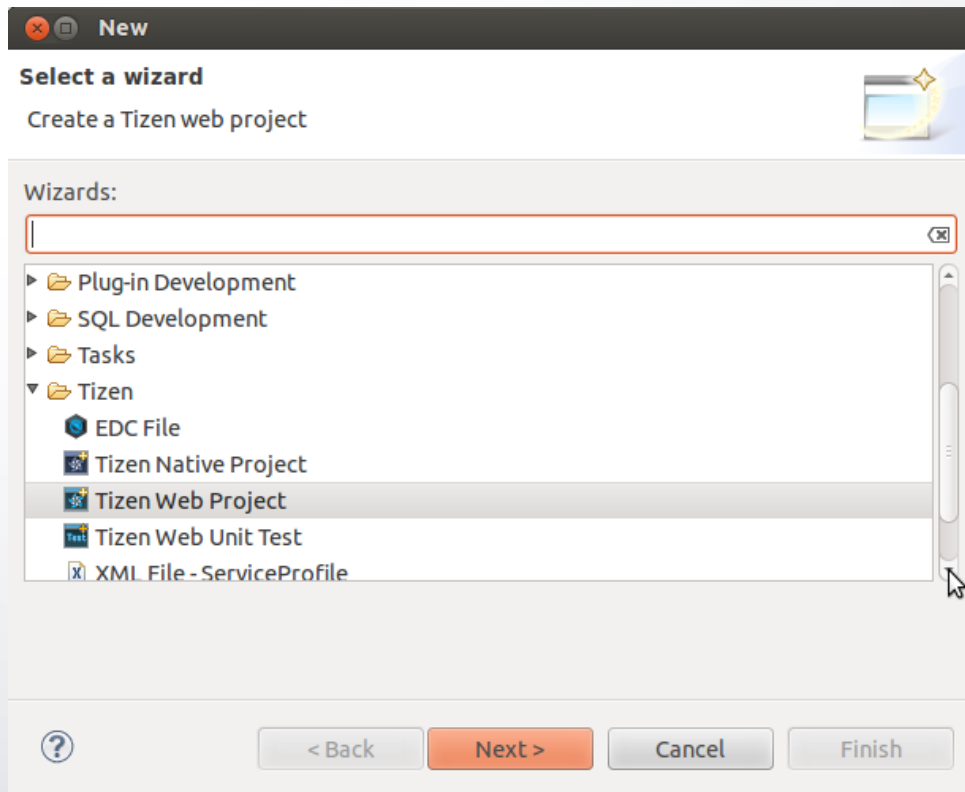


HTML5 Apps can do many things as well as native Apps.

Create a Web Project in SDK (1/2)

1. Create a web project

New -> Project

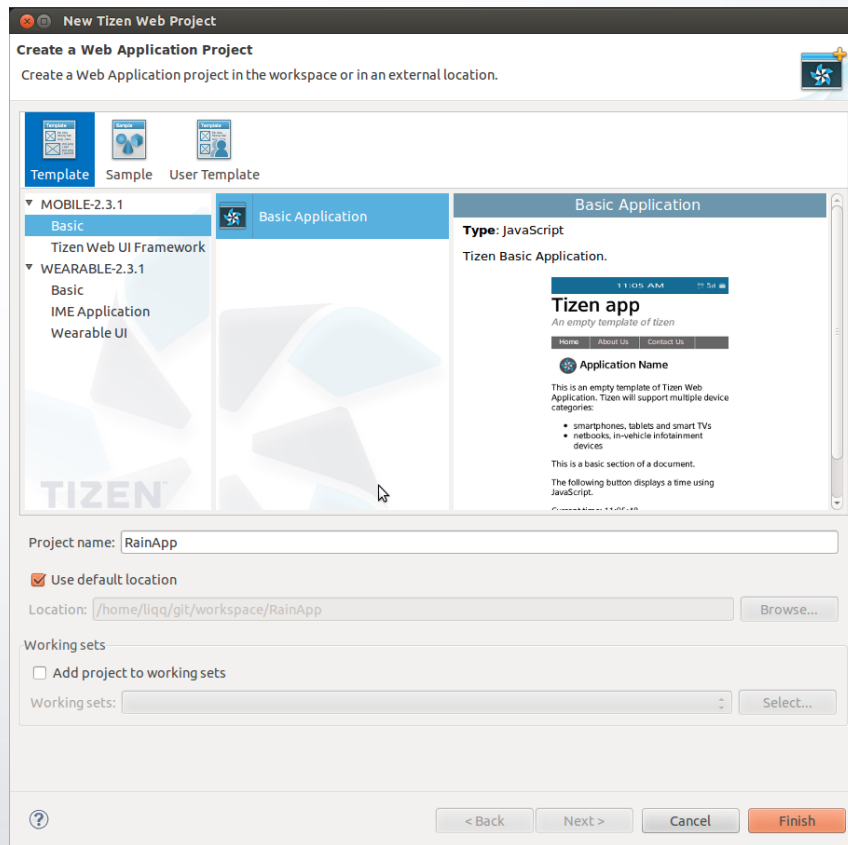


Create a Web Project in SDK (2/2)

2. Select a template, and input Project Name.

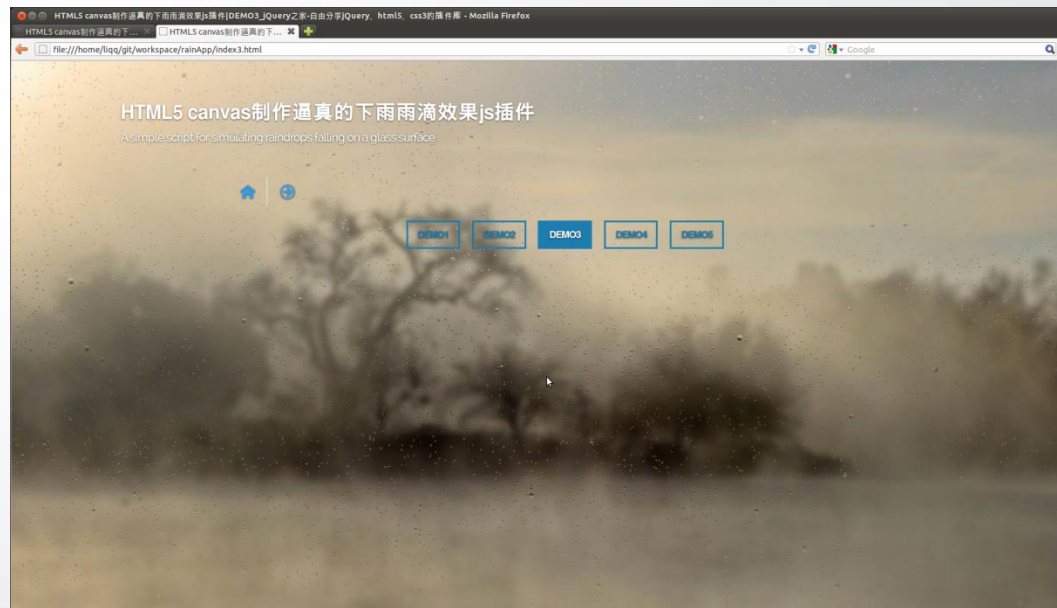
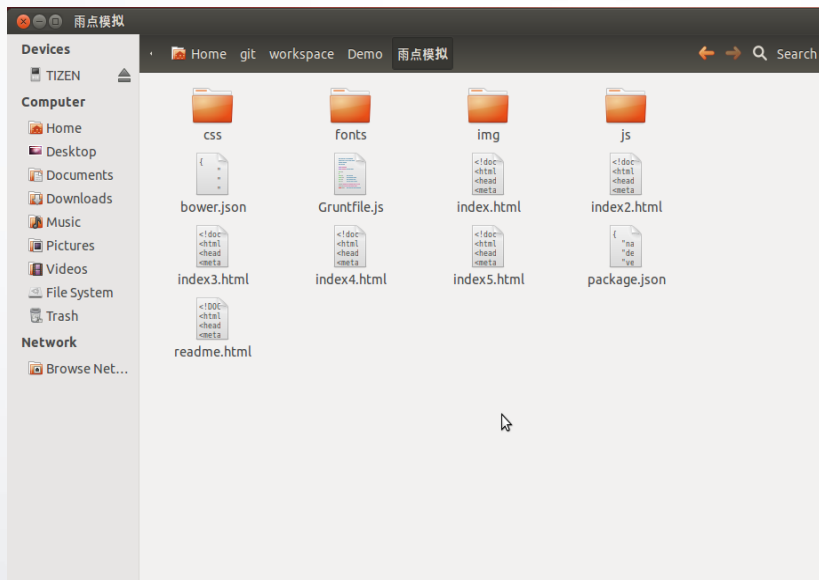
2.1 Mobile

2.2 Wearable



Copy Web App to SDK Project (1/3)

1. A web app sample download from internet

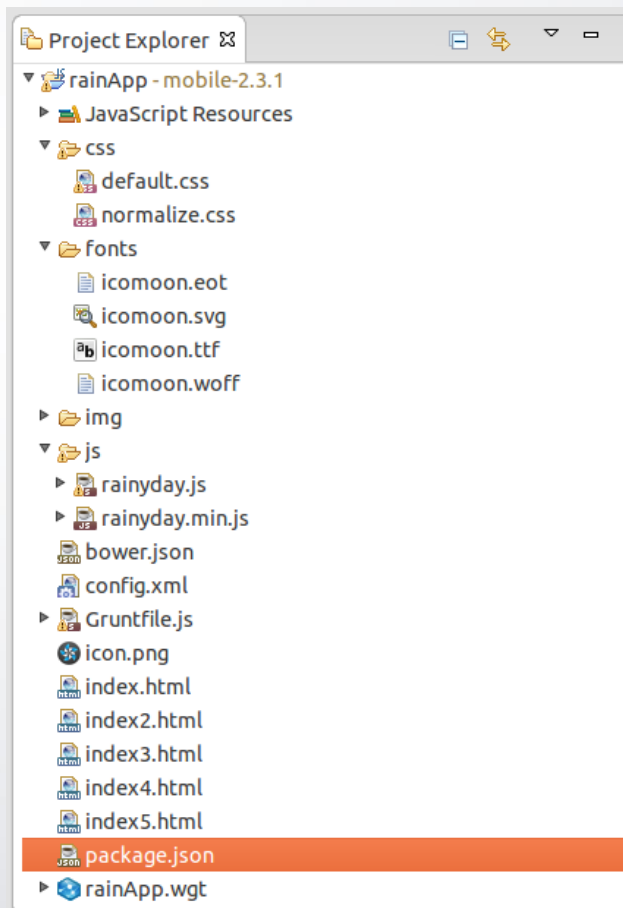


Copy Web App to SDK Project (2/3)

2. SDK Project Directory

Folder: css, js, images

File: config.xml, index.html

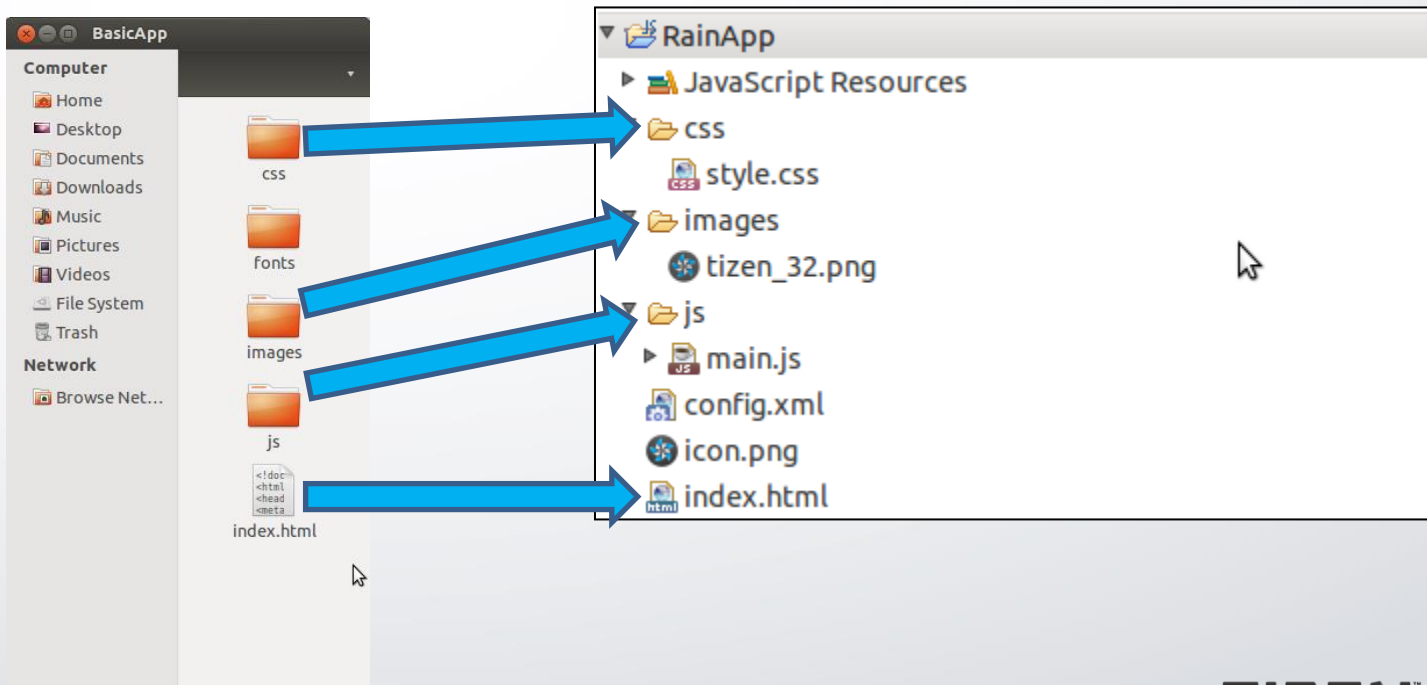


Copy Web App to SDK Project (3/3)

3. Copy files

Same name folder: copy files directly

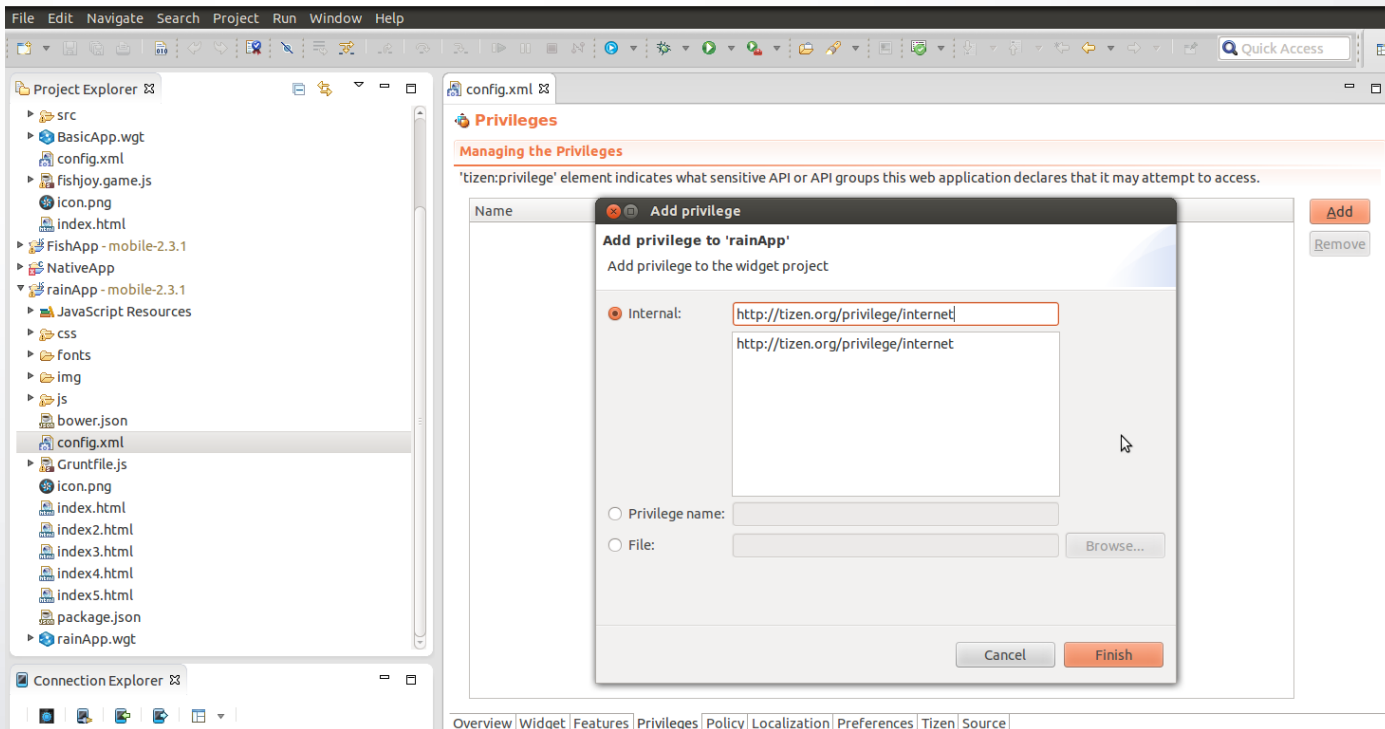
Otherwise: create new folder, and then copy files



Modify configure file (1/3)

1. Configure privilege


Example: Network permission



Modify configure file (2/3)

2. Configure Policy

Example: Network permission

 **Policy**

Managing the Policy

This section describes the URL navigation and content security policy for web application.
For more detailed description and usage of each field, please refer to "Tizen Web App Programming > Application Development Process > Setting Widget Configuration > Policy" guideline.

content-security-policy

content-security-policy-report-only

allow-navigation

access

Network URL	Allow subdomain
*	true

Add

Remove

Overview

Widget

Features

Privileges

Policy

Localization

Preferences

Tizen

Source

Modify configure file (3/3)

3. Configure source code

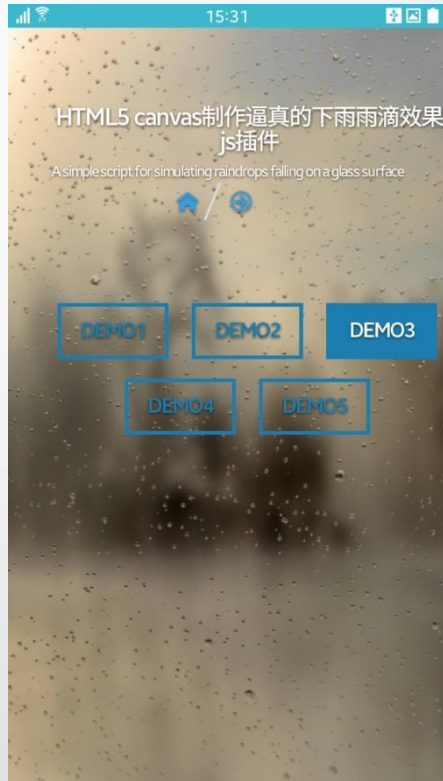
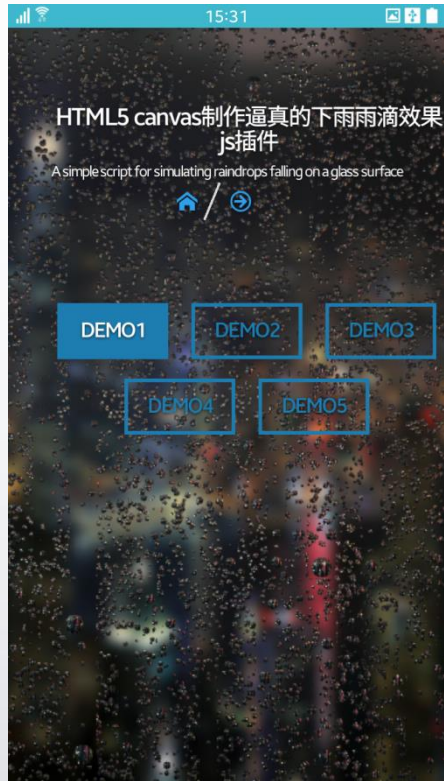
Example: Network permission

A screenshot of a code editor window titled 'config.xml'. The editor shows XML code for a Tizen application. The code includes a widget declaration with various attributes like xmlns, id, access, application, content, feature, icon, name, privilege, and profile. The code is color-coded: XML tags are in blue, attributes and values are in black, and string literals are in red. The code is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<widget xmlns="http://www.w3.org/ns/widgets" xmlns:tizen="http://tizen.org/ns/widgets" id="http://yourdomain/rainApp"
  <access origin="*" subdomains="true"></access>
  <tizen:application id="YINVXjrf6r.rainApp" package="YINVXjrf6r" required_version="2.3.1"/>
  <content src="index.html"/>
  <feature name="http://tizen.org/feature/screen.size.all"/>
  <icon src="icon.png"/>
  <name>rainApp</name>
  <tizen:privilege name="http://tizen.org/privilege/internet"/>
  <tizen:profile name="mobile"/>
</widget>
```

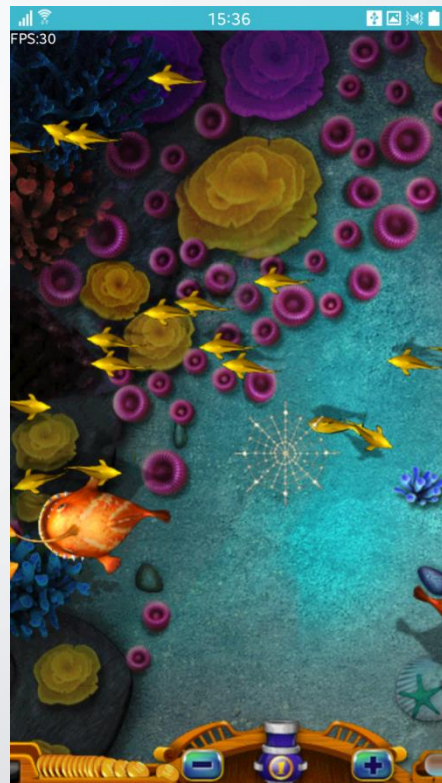
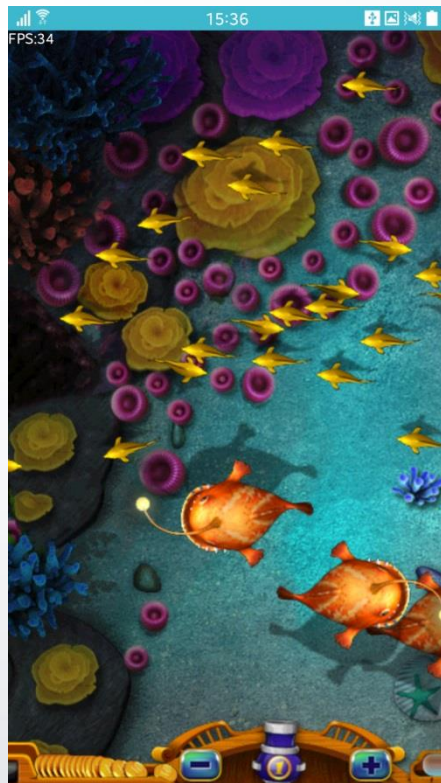
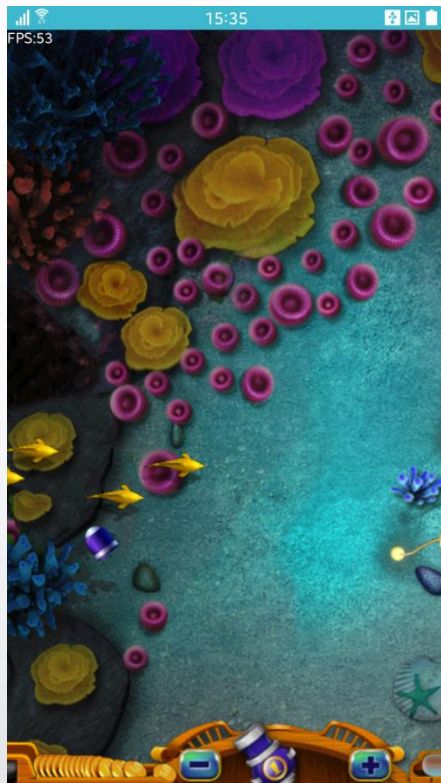
Run project in Tizen Target (1/2)

1. Run in the Tizen target
Run As -> Tizen Web Application



Run project in Tizen Target (2/2)

2. Launch app in target





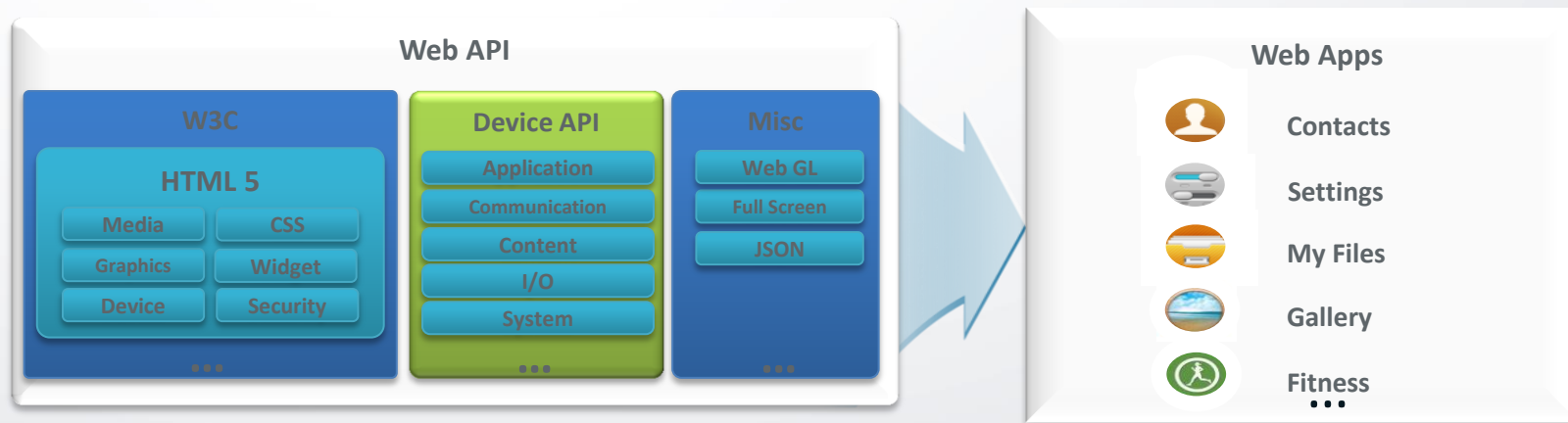
Tizen Web Device APIs

Content

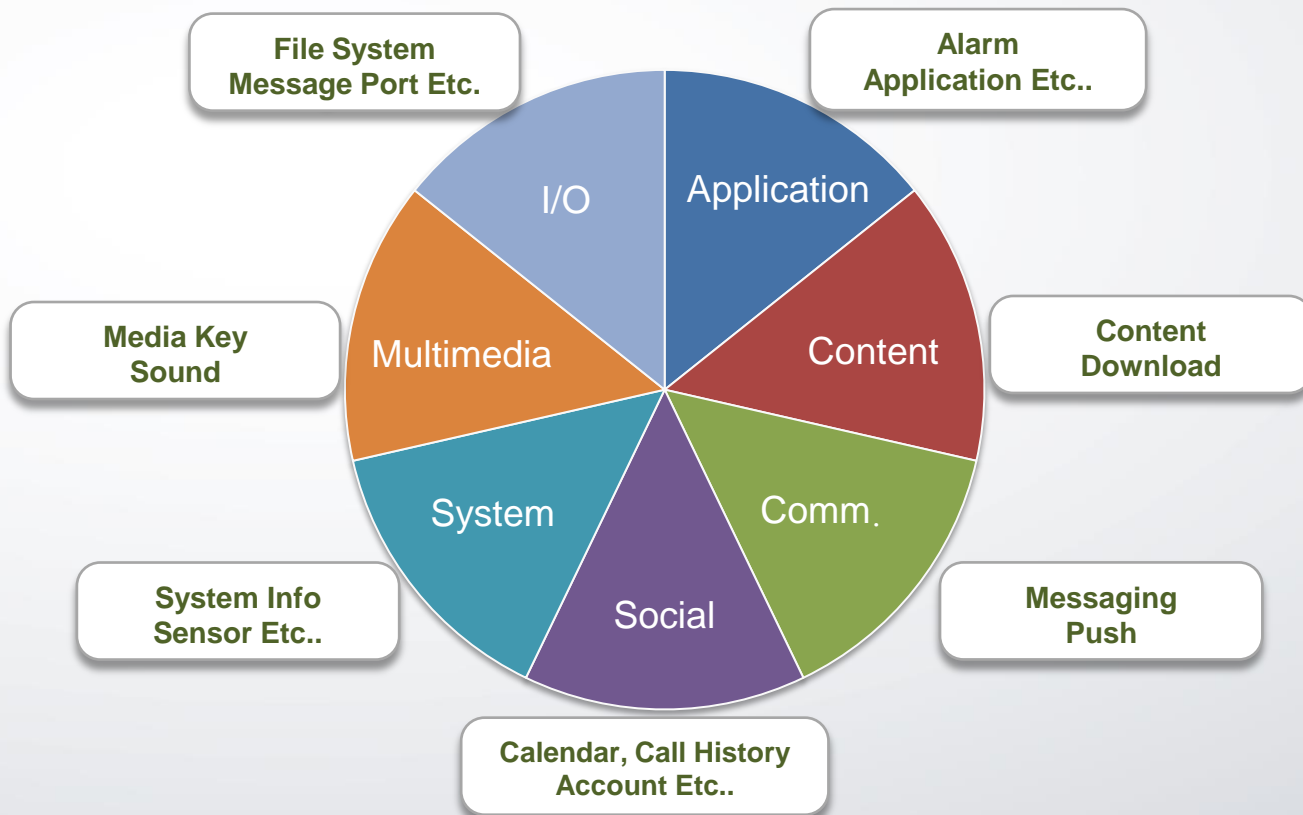
1. Overview
2. Privilege
3. Help content introduction
4. Use Bluetooth Device API

Overview (1/2) Tizen Web Device APIs

1. Apps need access to device features outside of the HTML
2. Can access various device features via Web Device API
3. Apps to work like native apps with minimum porting effort



Overview (2/2) Tizen Device APIs



Privilege

Optional – defines the API access privilege required by the Web App

Device API specific privilege setting

```
<?xml version="1.0" encoding="UTF-8"?>
<widget xmlns="http://www.w3.org/ns/widgets"
  xmlns:tizen="http://tizen.org/ns/widgets"
  id="http://yourdomain/BtSample"
  version="2.2.0" viewmodes="maximized">
  <tizen:application id="b9zxN75unx.BtSample" package="b9zxN75unx"
    required_version="2.2"/> <content src="index.html"/>
```

```
<tizen:privilege name="http://tizen.org/privilege/application.launch"/>
<tizen:privilege name="http://tizen.org/privilege/content.read"/>
<tizen:privilege name="http://tizen.org/privilege/content.write"/>
<tizen:privilege name="http://tizen.org/privilege/filesystem.read"/>
```

```
<tizen:privilege name="http://tizen.org/privilege/filesystem.read"/>
```

All privileges are described @<https://www.tizen.org/privilege>

```
</widget>
```

Help content introduction (1/2)

Help guide online: <https://developer.tizen.org>

Other Tizen Sites ▾

NewsEnglish ▾LoginRegister

≡ **TIZEN** Developers | DesignDevelopmentDistributionCommunity

Getting StartedTutorialsGuidesAPI ReferencesSampleToolsPreview

Web Application | Native Application

▼ Tizen Web Device API Reference

- ▼ Mobile Web
 - ▶ Tizen
 - ▶ Application
 - ▶ Communication
 - ▶ Content
 - ▶ Input/Output
 - ▶ Multimedia
 - ▶ Social
 - ▶ System
 - ▶ User Interface
- ▼ Wearable Web
 - ▶ Tizen
 - ▶ Application
 - ▶ Content
 - ▶ Input/Output
 - ▶ Multimedia
 - ▶ System
 - ▶ User Interface
- ▶ Tizen Web UI Framework Reference
- ▶ W3C/HTML5 and Supplementaries API Reference

[Tizen Web Device API Reference](#)

Tizen Mobile Web Device API Reference

The Tizen Web Device API, based on JavaScript, provides you advanced access to the device's platform capabilities.

You can develop rich Web applications using the Tizen Web Device APIs. You can, for example, control the application life-cycle, manage your schedules, exchange data, or make payments using NFC.

The APIs listed in this category are created by the Tizen platform to expose device capabilities to Web Applications.

Tizen Common

API	Description	Version (Since)	Mobile	Supported on Mobile Emulator
Tizen	The base object for accessing the Tizen Web Device API.	1.0	Mandatory	Yes

Application

API	Description	Version (Since)	Mobile	Supported on Mobile Emulator
Alarm	This API provides functionality for setting and unsetting alarms.	1.0	Mandatory	Yes
Application	This API provides information about running and installed applications and controls them.	1.0	Mandatory	Yes
Data Control	This API provides interfaces and methods for accessing specific data exported by other applications.	2.1	Mandatory	Yes
Package	This API provides information install/uninstall package and get information about installed packages.	2.1	Mandatory	Yes

TIZEN™ DEVELOPER
CONFERENCE
2015 SHENZHEN

26

Help content introduction (2/2)

Help content on SDK

The screenshot shows the Tizen IDE Help window. The left sidebar contains a 'Contents' tree with the following structure:

- Getting Started with Tizen
- Tools
- Tutorials
- Guides
- UI Guides
- API References
 - Web Application
 - Tizen Web Device API Reference
 - Mobile Web
 - Tizen
 - Application
 - Communication
 - Bluetooth**
 - Messaging
 - Network Bearer Selection
 - NFC
 - Push
 - Secure Element
 - Content
 - Input/Output
 - Multimedia
 - Social
 - System
 - User Interface
 - Wearable Web
 - Tizen Advanced UI framework (TAU)
 - W3C/HTML5 and Supplimentary API Reference
 - Native Application
 - Sample Descriptions
 - Eclipse Development Documentation

The main content area displays the 'Bluetooth API' documentation. At the top, the breadcrumb path is: [API References](#) > [Web Application](#) > [Tizen Web Device API Reference](#) > [Mobile Web](#) > [Communication](#).

Bluetooth API

The Bluetooth API defines interfaces and methods to manage Bluetooth.

The following Bluetooth functionalities are provided:

- Controls local Bluetooth device, that is, turn Bluetooth on/off, etc.
- Sets visibility
- Discovers nearby Bluetooth devices (Device discovery, including Bluetooth LE devices)
- Gets bonded devices information
- Controls bonding
- Connects to a service on a remote device and exchanges data
- Registers a service (RFCOMM) on a local device, which can be consumed by remote devices to exchange data
- Advertise for remote devices (including Bluetooth LE devices)
- Act as a GATT client (Generic Attribute Profile client)

For more information on the Bluetooth features, see [Bluetooth Guide](#).

Since: 1.0

Table of Contents

- Type Definitions
 - [BluetoothAddress](#)
 - [BluetoothUUID](#)
 - [BluetoothSocketState](#)
 - [BluetoothProfileType](#)
 - [BluetoothHealthChannelType](#)
 - [BluetoothESolicitationUUID](#)
 - [BluetoothAdvertisePacketType](#)
 - [BluetoothAdvertisingState](#)
 - [BluetoothAdvertisingMode](#)
- Interfaces
 - [BluetoothManagerObject](#)
 - [BluetoothLEServiceData](#)
 - [BluetoothLEManufacturerData](#)
 - [BluetoothLEAdvertiseDataInit](#)
 - [BluetoothLEAdvertiseData](#)
 - [BluetoothManager](#)
 - [BluetoothAdapter](#)
 - [BluetoothLEAdapter](#)
 - [BluetoothGATTService](#)
 - [BluetoothGATTCharacteristic](#)
 - [BluetoothGATTDescriptor](#)
 - [BluetoothLEScanCallback](#)
 - [BluetoothLEAdvertiseCallback](#)
 - [BluetoothLEConnectChangeCallback](#)
 - [ReadValueSuccessCallback](#)
 - [BluetoothDevice](#)
 - [BluetoothLEDevice](#)
 - [BluetoothSocket](#)

The status bar at the bottom shows the URL: http://127.0.0.1:37537/help/topic/org.tizen.web.apireference/html/device_api/device_api_cover.html.

Use Bluetooth Device API (1/6)

Search Bluetooth target & Connect it



Use Bluetooth Device API (2/6)

Add Privilege

```
<?xml version="1.0" encoding="UTF-8"?>
<widget xmlns="http://www.w3.org/ns/widgets" xmlns:tizen="http://tizen.org/ns/widgets" id="http://yo
  <tizen:application id="XubhEoac7s.BluetoothChat" package="XubhEoac7s" required_version="2.3"/>
  <content src="index.html"/>
  <feature name="http://tizen.org/feature/screen.size.all"/>
  <icon src="icon.png"/>
  <name>BluetoothChat</name>
  <tizen:privilege name="http://tizen.org/privilege/application.launch"/>
  <tizen:privilege name="http://tizen.org/privilege/bluetooth.admin"/>
  <tizen:privilege name="http://tizen.org/privilege/bluetooth.gap"/>
  <tizen:privilege name="http://tizen.org/privilege/bluetooth.spp"/>
  <tizen:profile name="mobile"/>
  <tizen:setting screen-orientation="portrait" context-menu="disable" background-support="disable"
</widget>
```

Use Bluetooth Device API (3/6)

Use Application APIs to control App lifecycle

```
window.onload = function () {  
    // TODO:: Do your initialization job  
  
    // add eventListener for tizenhwkey  
    document.addEventListener('tizenhwkey', function(e) {  
        if(e.keyName == "back")  
        try {  
            tizen.application.getCurrentApplication().exit();  
        } catch (ignore) {  
        }  
    });  
};
```

Use Bluetooth Device API (4/6)

Add BT Adapter, Register Search Callback

```
var adapter = tizen.bluetooth.getDefaultAdapter();
function startDiscovery()
{
    var discoverDevicesSuccessCallback =
    {
        onstarted: function()
        {
            console.log("Device discovery started...");
        },
        ondevicefound: function(device)
        {
            console.log("Found device - name : " + device.name + ", Address : " + device.address);
        },
        ondevice disappeared: function(address)
        {
            console.log("Device disappeared : " + address);
        },
        onfinished: function(devices)
        {
            //find the device
            console.log("discover finished");
            var message;
            $("#btlist").html("");
            var listr = "";
            for (var i=0;i<devices.length; i++)
            {
                message = "devices Name : " + devices[i].name + ", Address: " + devices[i].address + "\n";
                console.log(message);
                devicesAddr.push(devices[i].address);
                listr = "<li>" + devices[i].name + "</li>";
                $("#btlist").append(listr);
            }
            message += "Total : " + devices.length;
            console.log(message);

            if (devices.length > 0) {
                $("#btlist").children("li").bind("click", function(){
                    var $thisli = $(this);
                    selectIndex = $thisli.index();
                    console.log("li index:" + selectIndex);
                });
            }
        }
    };
    // start searching for nearby devices, for 12 seconds
    console.log("discoverDevices");
    adapter.discoverDevices(discoverDevicesSuccessCallback, function(e){
        console.log("Failed to search devices : " + e.message + "(" + e.name + ")");
    });
}
```

```
var adapter = tizen.bluetooth.getDefaultAdapter();
function startDiscovery()
{
    var discoverDevicesSuccessCallback =
```

```
$("#btlist").html("");
var listr = "";
for (var i=0;i<devices.length; i++)
{
    message = "devices Name : " + devices[i].name + ", Address: " + devices[i].address + "\n";
    console.log(message);
    devicesAddr.push(devices[i].address);
    listr = "<li>" + devices[i].name + "</li>";
    $("#btlist").append(listr);
}
message += "Total : " + devices.length;
console.log(message);
```

```
adapter.discoverDevices(discoverDevicesSuccessCallback, function(e){
    console.log("Failed to search devices : " + e.message + "(" + e.name + ")");
});
```

Use Bluetooth Device API (5/6)

Start to search targets

```
function onSetPoweredError(e)
{
    console.log("Could not turn on device, reason : " + e.message + "(" + e.name + ")");
}
try
{
    console.log("setPowered");
    adapter.setPowered(true, startDiscovery, onSetPoweredError);
}
catch(e)
{
    console.log(e.message);
}
```

Click connect button to pair target

```
$("#conn_btn").click(function(){
    $("#statuslist").html('');
    var message = "start to connect.";
    printfMessage(message);

    alert(selectIndex);
    paireDevice(selectIndex);
});
```


A Bluetooth Sample (6/6)

Pair target & Show it

```
function paireDevice(index)
{
    var i;
    function onBondingSuccess(device)
    {
        pairedDevice = device;
        console.log("Device Name :" + device.name);
        console.log("Device Address :" + device.address);

        for (i=0;i<device.uuids.length; i++) {
            console.log("[ "+i+" ]" + "Device Service UUIDs :" + device.uuids[i]);
        }

        //console.log("Device Service UUIDs :" + device.uuids); //device.uuids.join("\n")

        if (pairedDevice.uuids.indexOf("5BCE9431-6C75-32AB-AFE0-2EC108A30866") != -1) {
            console.log("-----connectToServiceByUUID-----");
            pairedDevice.connectToServiceByUUID("5BCE9431-6C75-32AB-AFE0-2EC108A30866", onSocketConnected,
            console.log ("Error connecting to service. Reason: " + e.message);
            });
        }

        var message = "pared device name: " + device.name + "success !!!";
        printfMessage(message);
    }
    function onError(e)
    {
        console.log ("Could not create bonding, reason :" + e.message);
    }
    var adapter = tizen.bluetooth.getDefaultAdapter();
    adapter.createBonding(devicesAddr[index], onBondingSuccess, onError);
}
```

**var message = "pared device name: " + device.name + "success !!!";
printfMessage(message);**

**var adapter = tizen.bluetooth.getDefaultAdapter();
adapter.createBonding(devicesAddr[index], onBondingSuccess, onError);**

An abstract graphic on the left side of the slide. It features several triangles of varying sizes, some with blue and white horizontal stripes and others with solid light blue. A trail of small, colorful dots (blue, green, and dark blue) extends from the bottom of the striped triangles towards the bottom right corner of the slide.

TAU Introduction

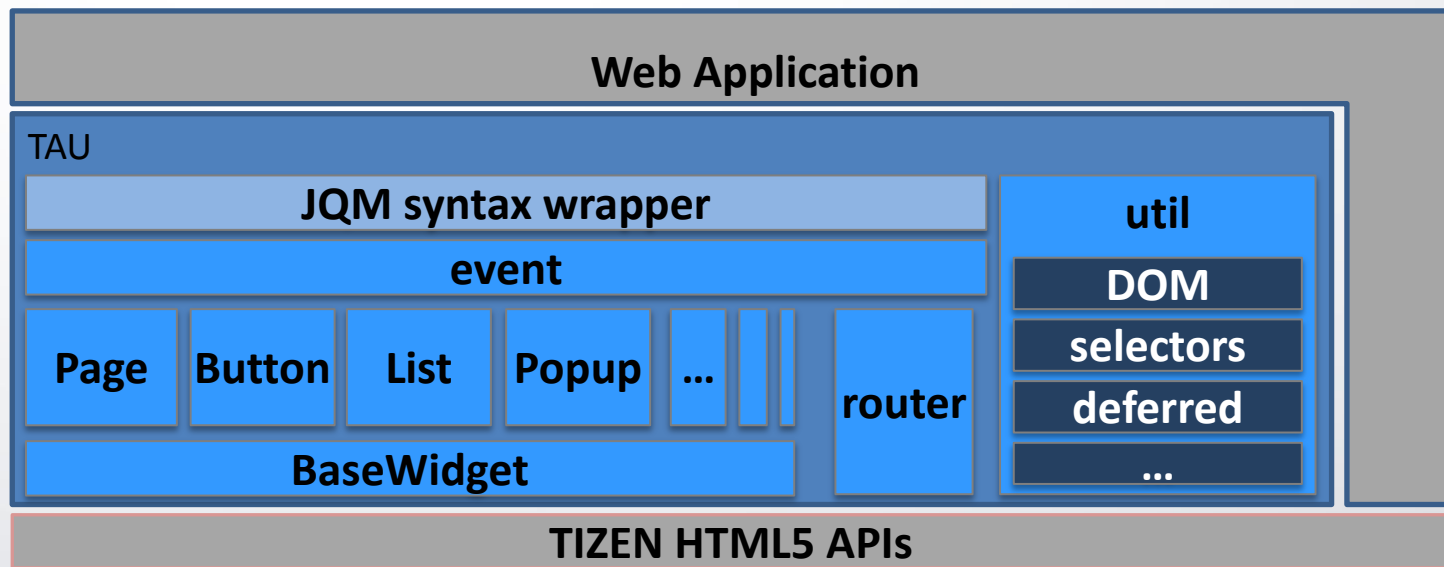
Content

1. Overview
2. Page Structure
3. TAU Components

Overview (1/2)

Tizen Advanced UI framework

A collection of library providing TIZEN UI for web apps, consists of CSS and Javascript



Overview (2/2)

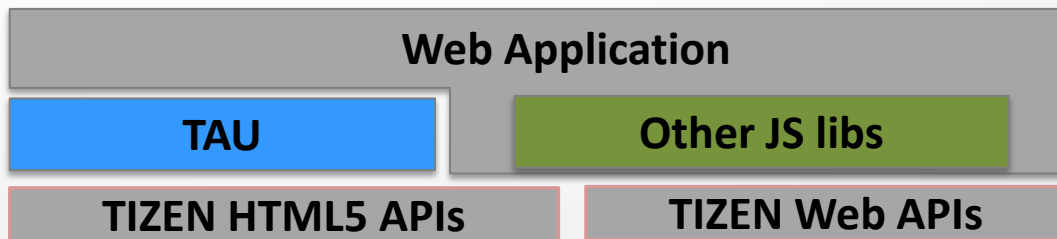
Tizen provides most HTML5 specs

TAU is **optional** to web apps

Mainly for using Tizen look&feel UI controls

Typical header/content/footer style mobile/wearable/TV apps

Useful for the apps who wants to use Tizen UI



Page Structure (1/4)

1. Header

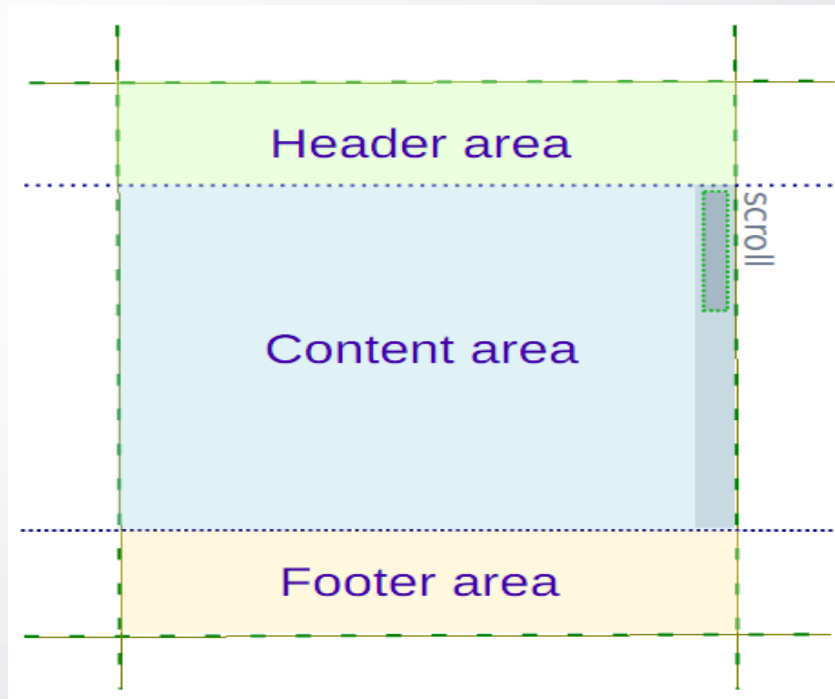
best place to mark out to user
what page is currently opened,
it can contain buttons, menus and
toolbars (optional)

2. Content

where the main application
content resides, it has an
optional scroll bar

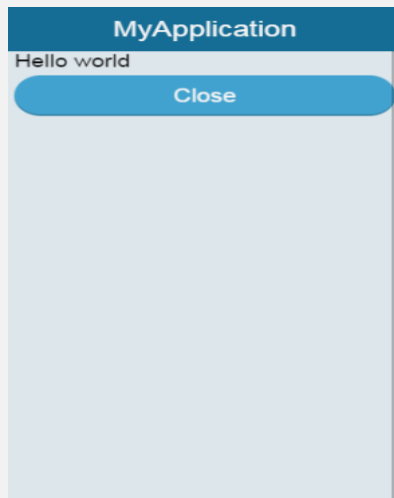
3. Footer

here you can put a status line or
buttons (optional)

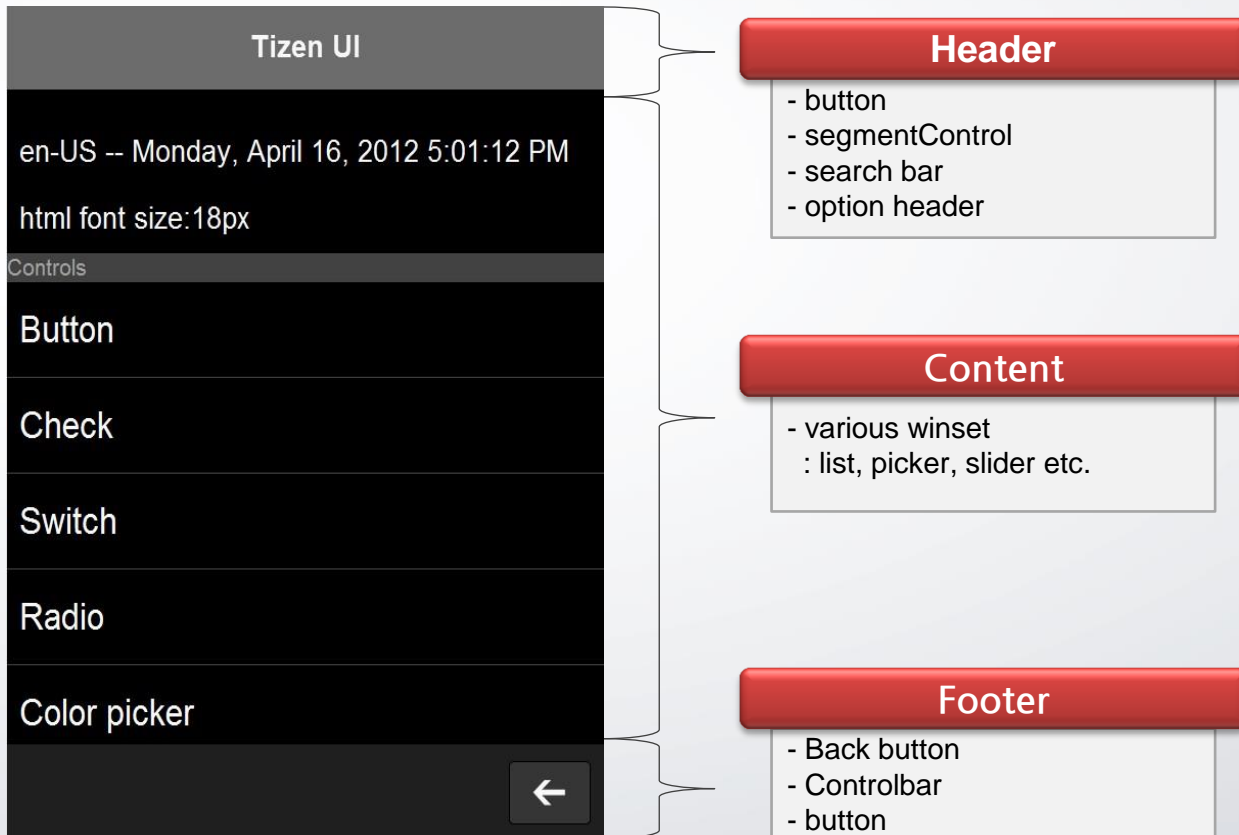


Page Structure (2/4)

```
<!DOCTYPE html>
<html>
<head>
  <meta name="viewport" content="width=device-width,user-scalable=no">
  <link rel="stylesheet" href="lib/tau/mobile/theme/default/tau.min.css">
  <script src="lib/tau/mobile/js/tau.min.js"></script>
</head>
<body>
  <div data-role="page">
    <div data-role="header">
      <h1>MyApplication</h1>
    </div>
    <div data-role="content">
      Hello world
      <button id="closeBtn">Close</button>
    </div>
  </div>
</body>
</html>
```



Page Structure (3/4)



Page Structure (4/4)

Change Page in TAU

1. Use

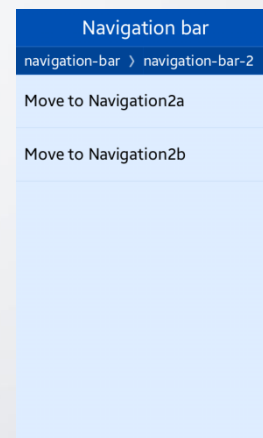
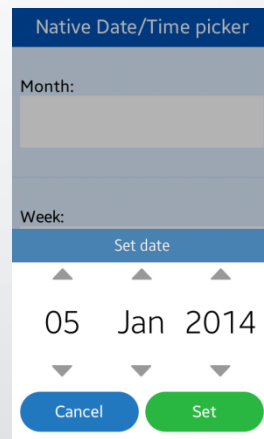
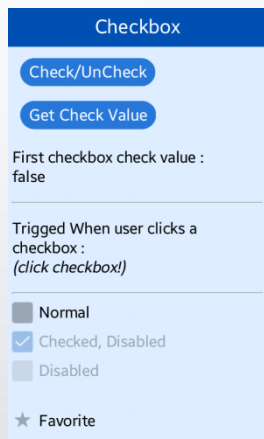
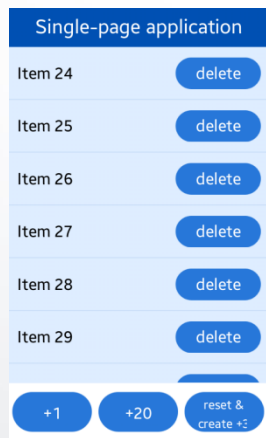
```
<a href="#page2"><button>Move to Page2</button></a>
```

2. Use tau.changePage()

```
<button id="btn">Move to Page2</button>
<script>
  var button=document.querySelector("#btn");
  button.addEventListener("click", function() {
    tau.changePage("#page2");
  });
</script>
```

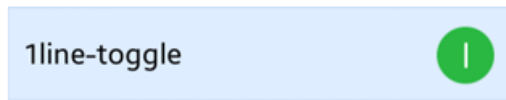
```
<div class="ui-page" id="page2">
  <div id="timer-up-total-time">00:00:00</div>
</div>
```

TAU Components (1/4)



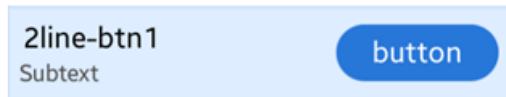
TAU Components (2/4)

– Dialogue → 1line toggle



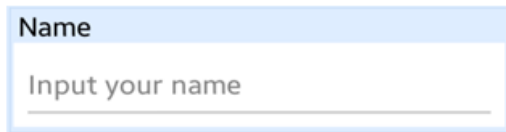
```
<li class="ui-li-dialogue">
  1line-toggle
  <select name="flip-11" id="flip-11" data-role="slider">
    <option value="off"></option>
    <option value="on"></option>
  </select>
</li>
```

– Dialogue → 2line-btn1



```
<li class="ui-li-has-multiline ui-li-dialogue ui-li-text-ellipsis">
  <a>
    2line-btn1 (with link)
    <span class="ui-li-text-sub">Subtext</span>
    <div data-role="button" data-inline="true">button</div>
  </a>
</li>
```

– Dialogue (edit mode) → Name



```
<li class="ui-li-dialogue ui-li-dialogue-edit">
  <div class="ui-edit-title">Name </div>
  <input placeholder="Input your name" />
</li>
```

TAU Components (3/4)

- Popup → Popup with 1line title

Popup title

Pop-up dialog box, a child window that blocks user interact to the parent windows

```
<div id="center_title" data-role="popup" class="center_title">  
  <div class="ui-popup-title">  
    <h1>Popup title</h1>  
  </div>  
  <div class="ui-popup-text">  
    Pop-up dialog box, a child  
    window that blocks user interact  
    to the parent windows  
  </div>  
</div>
```

- Popup → Popup with 2buttons

Popup title

Pop-up dialog box, a child window that blocks user interact to the parent windows

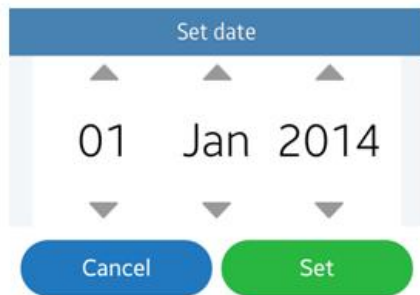
Button1

Button2

```
<div id="center_title_2btn" data-role="popup" class="center_title_2btn">  
  <div class="ui-popup-title">  
    <h1>Popup title</h1>  
  </div>  
  <div class="ui-popup-text">  
    Pop-up dialog box, a child  
    window that blocks user interact  
    to the parent windows  
  </div>  
  <div class="ui-popup-button-bg">  
    <a data-role="button" data-rel="back" data-inline="true">Button1</a>  
    <a data-role="button" data-rel="back" data-inline="true">Button2</a>  
  </div>  
</div>
```

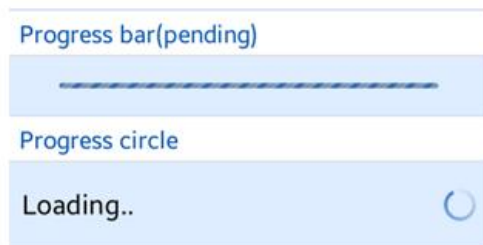
TAU Components (4/4)

Datetime Picker



```
<li><div data-role="fieldcontain">
  <label for="month">Month:</label>
  <input type="month" name="month" id="month" value="" />
</div></li>
```

Progress bar



```
<li data-role="list-divider">Progress bar(pending)</li>
<li id="pendingTest" >
  <div data-role="progress" data-style="pending" id="pending"></div>
</li>

<li data-role="list-divider">Progress circle</li>
<li id="progressingTest" >
  <div data-role="progress" data-style="circle" id="progressing"></div>
  Loading..
</li>
```

End

Thank You