

OpenWebOS WebApp

May, 2016
Seungjae Baek

Dept. of software
Dankook University

<http://embedded.dankook.ac.kr/~baeksj>

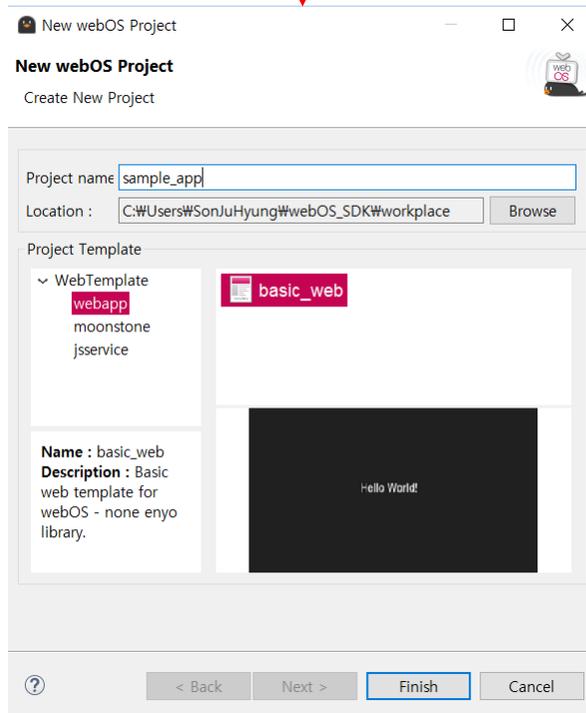
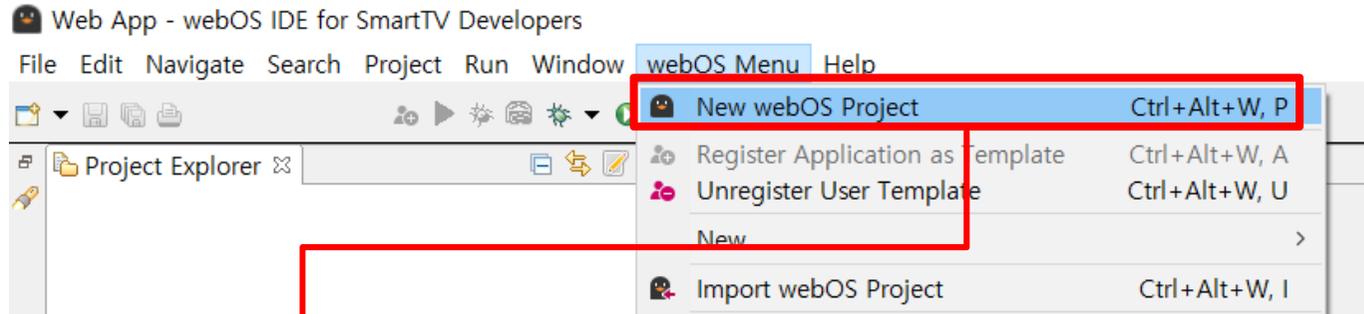
A vertical blue line with three solid blue circles, serving as a decorative element on the left side of the slide.

1. WebApp(WebStorage: Local Storage & Session Storage)

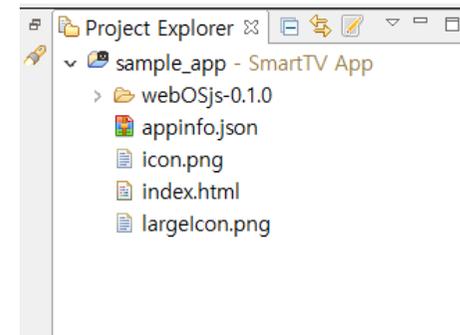
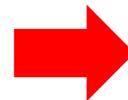
- ✓ Create WebApp project
- ✓ Edit CSS in HTML head
- ✓ Edit JavaScript

Development Process using HTML

■ Create a new Project

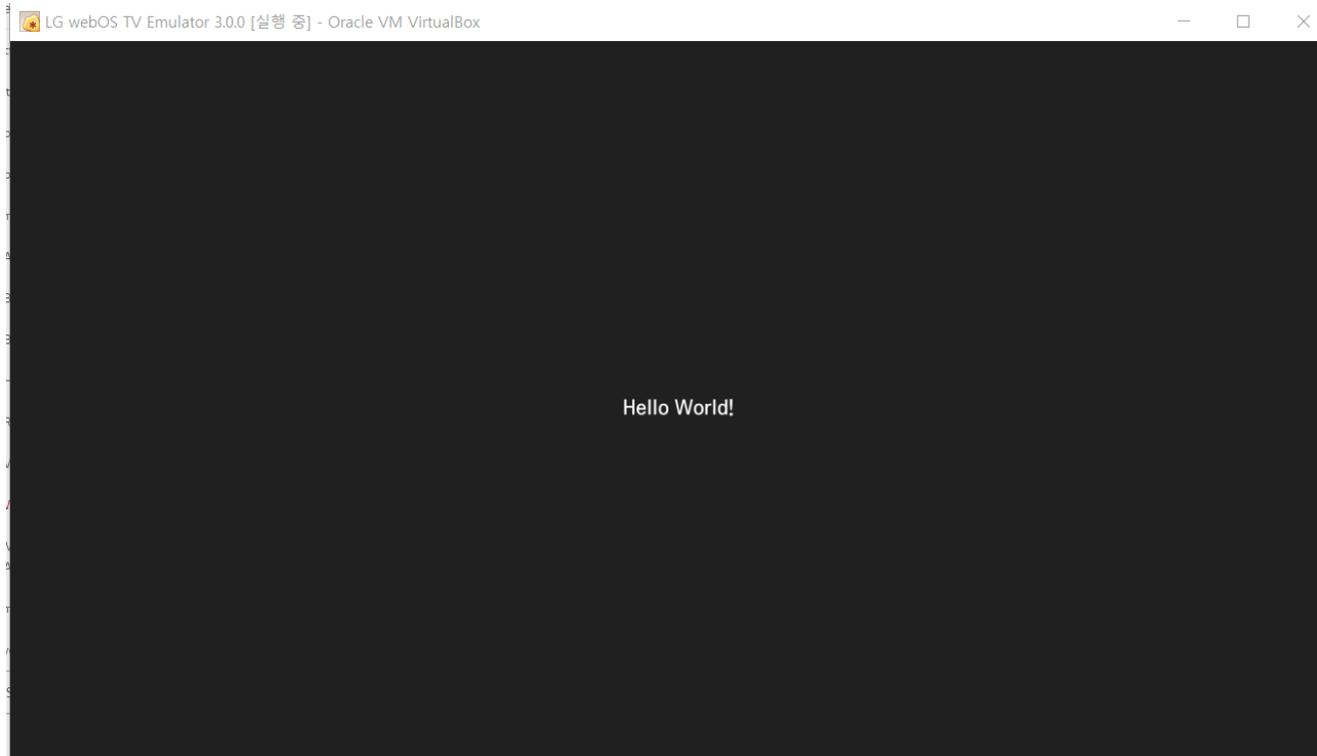


- ✓ webOS Menu -> New webOS Project
- ✓ Project name 입력
- ✓ wepapp 선택
- ✓ Finish



Development Process using HTML

■ Create a new Project



- ✓ Basic HelloWorld App
- ✓ Edit this project to create an app which accesses web storage

Development Process using HTML

- Edit CSS in HTML head

```
<style type="text/css">
body {
  font-family: "Miso", serif;
  width: 100%;
  height: 100%;
  background-color: #202020;
}
div {
  position: absolute;
  height: 100%;
  width: 100%;
  display: table;
  font-size: 60px;
  color: #ffffff;
}
lg_red {
  color: #d80546;
}
button {
  height: 80px;
  font-size: 60px;
}
</style>
```

Development Process using HTML

■ Edit JavaScript in HTML head

```
<script type="text/javascript" charset="utf-8">

function clickCounter() {
    if(typeof(Storage) !== "undefined") {

        // Check Local Storage
        if (localStorage.clickcount) {
            localStorage.clickcount = Number(localStorage.clickcount) + 1;
        } else {
            localStorage.clickcount = 1;
        }

        // Check Session Storage
        if (sessionStorage.clickcount) {
            sessionStorage.clickcount = Number(sessionStorage.clickcount) + 1;
        } else {
            sessionStorage.clickcount = 1;
        }
        document.getElementById("resultLocal").innerHTML = "LOCAL Storage: <lg_red>" + localStorage.clickcount + "</lg_red> time(s).";
        document.getElementById("resultSession").innerHTML = "SESSION Storage: <lg_red>" + sessionStorage.clickcount + "</lg_red> time(s).";
    } else {
        document.getElementById("resultLocal").innerHTML = "Sorry, your browser does not support web storage function.";
        document.getElementById("resultSession").innerHTML = "Sorry, your browser does not support web storage function.";
    }
}

</script>
```

Development Process using HTML

■ Edit JavaScript in HTML head

```
function clickCounter() {  
    if(typeof(Storage) !== "undefined") {  
  
        // Check Local Storage  
        if (localStorage.clickcount) {  
            localStorage.clickcount = Number(localStorage.clickcount) + 1;  
        } else {  
            localStorage.clickcount = 1;  
        }  
    }  
}
```

- ✓ clickCounter()
 - localStorage
 - Store value locally with local storage in webOS TV
 - Saves data permanently
 - Large amounts(at least 5MB) of data can be stored locally, without affecting website performance.
 - Check local storage and add count
 - Number() function converts the object argument to a number that represents the object's value.

Development Process using HTML

■ Edit JavaScript in HTML head

```
// Check Session Storage
if (sessionStorage.clickcount) {
    sessionStorage.clickcount = Number(sessionStorage.clickcount) + 1;
} else {
    sessionStorage.clickcount = 1;
}
```

- ✓ clickCounter()
 - sessionStorage
 - Saves value on browser only for the duration of while app is running
 - Check local storage and add count

Development Process using HTML

■ Edit JavaScript in HTML head

```
document.getElementById("resultLocal").innerHTML = "LOCAL Storage: <lg_red>" + localStorage.clickcount + "</lg_red> time(s).";  
document.getElementById("resultSession").innerHTML = "SESSION Storage: <lg_red>" + sessionStorage.clickcount + "</lg_red> time(s).";  
} else {  
document.getElementById("resultLocal").innerHTML = "Sorry, your browser does not support web storage function.";  
document.getElementById("resultSession").innerHTML = "Sorry, your browser does not support web storage function.";  
}  
}
```

- ✓ document.getElementById()
 - Returns the element that has the ID attribute with the specified value.
 - Returns *null* if no elements with the specified ID exists.
 - ID should be unique within a page.
 - If more than one element with the specified ID exists returns the first element in the source code.
- ✓ localStorage.clickcount , sessionStorage.clickcount
 - Show click count to screen.

Development Process using HTML

■ Edit JavaScript in HTML head

```
<body onload="clickCounter()">
  <div>
    <h1 id="resultLocal">Hello! World!</h1>
    <h1 id="resultSession">Hello! World!</h1>
    <button onclick="clickCounter()" type="button" style="font-family: 'Miso';"><b>Click Here to <lg_red>Add Count</b></lg_red>!</button>
  </div>
</body>
```

- ✓ Add button to show Add Count
- ✓ Add count result portion using h1 tag

Development Process

- Testing on Emulator





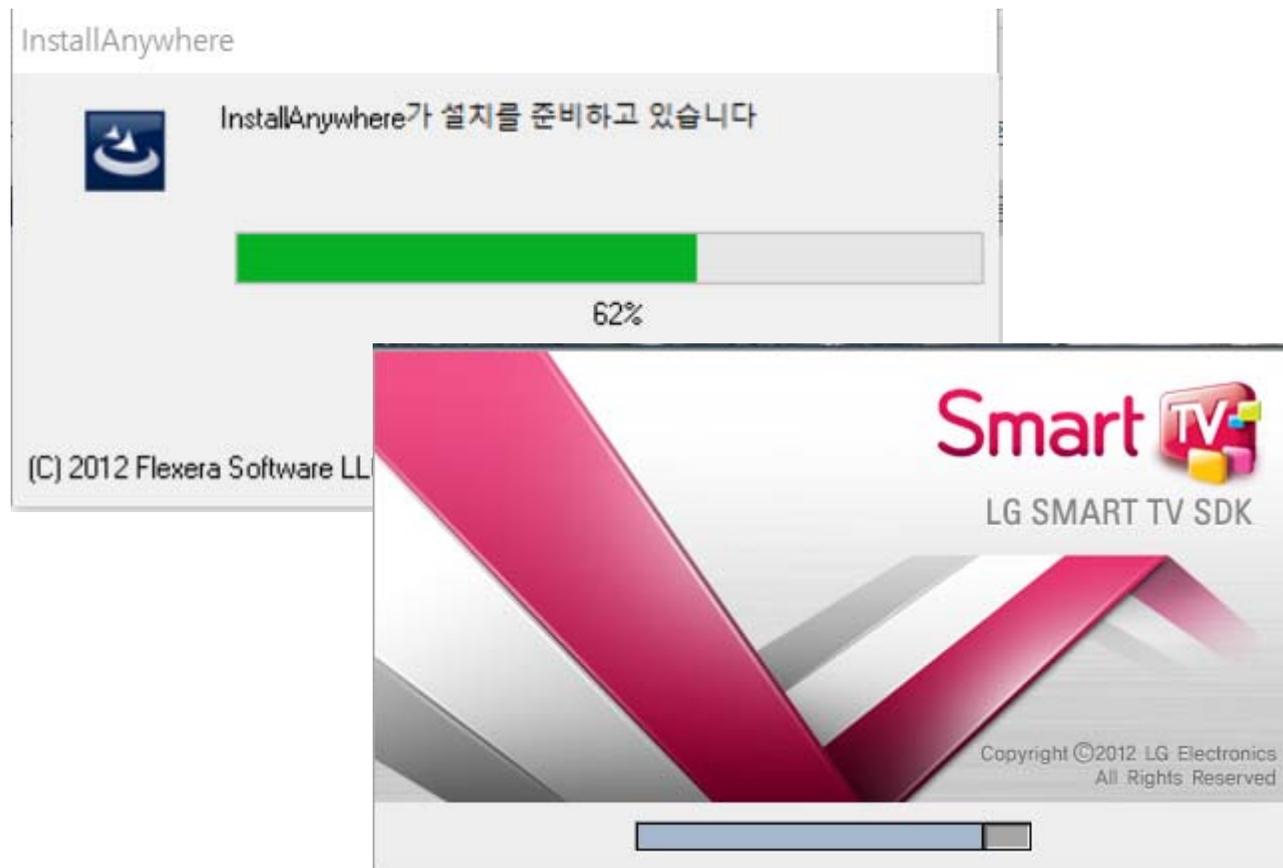
2. NetCastApp

- ✓ Install netcast SDK
- ✓ Create webApp Project
- ✓ Add CSS, JavaScript to project
- ✓ Implementation

Development Process

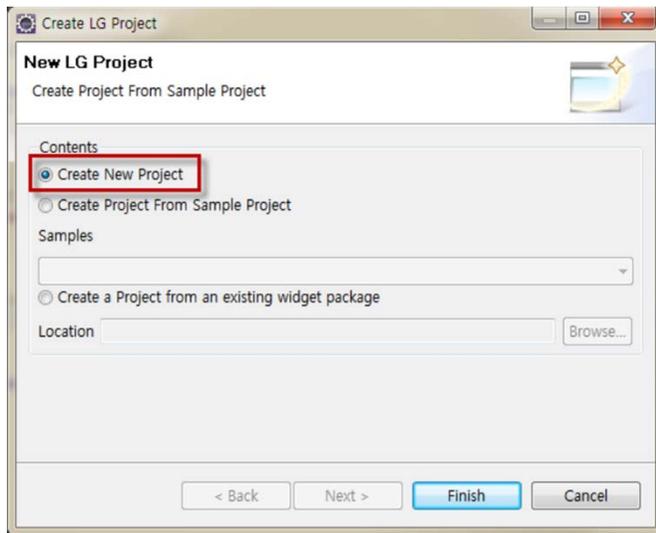
■ Installing SDK

- ✓ <http://developer.lge.com/resource/tv/RetrieveNetCastSdktoolsContent.dev#none>

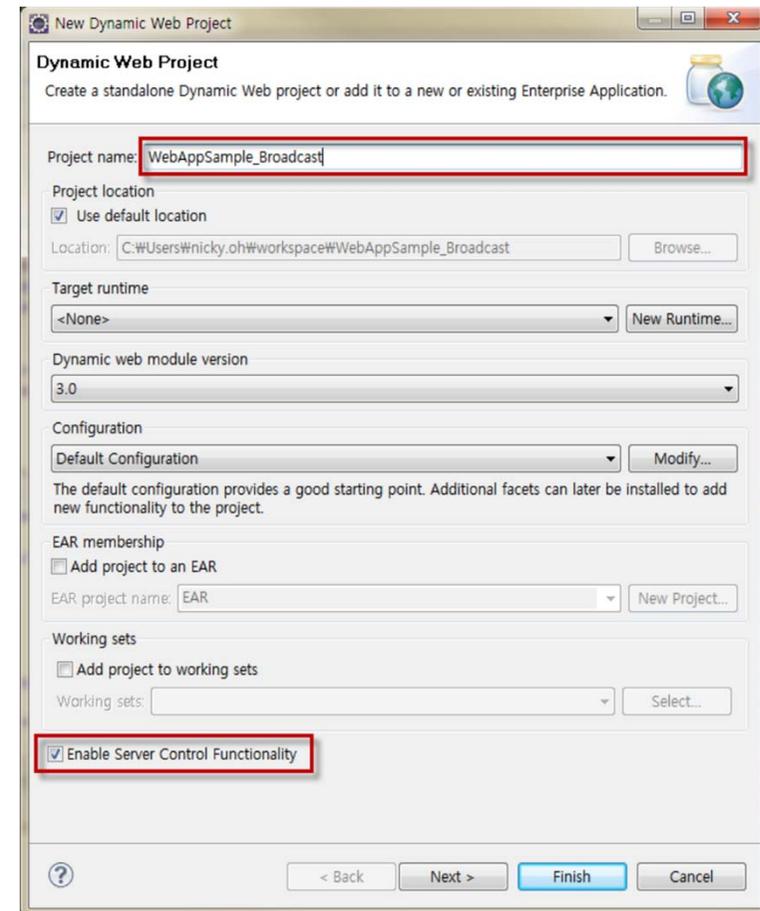


Development Process

■ Creating LG Project

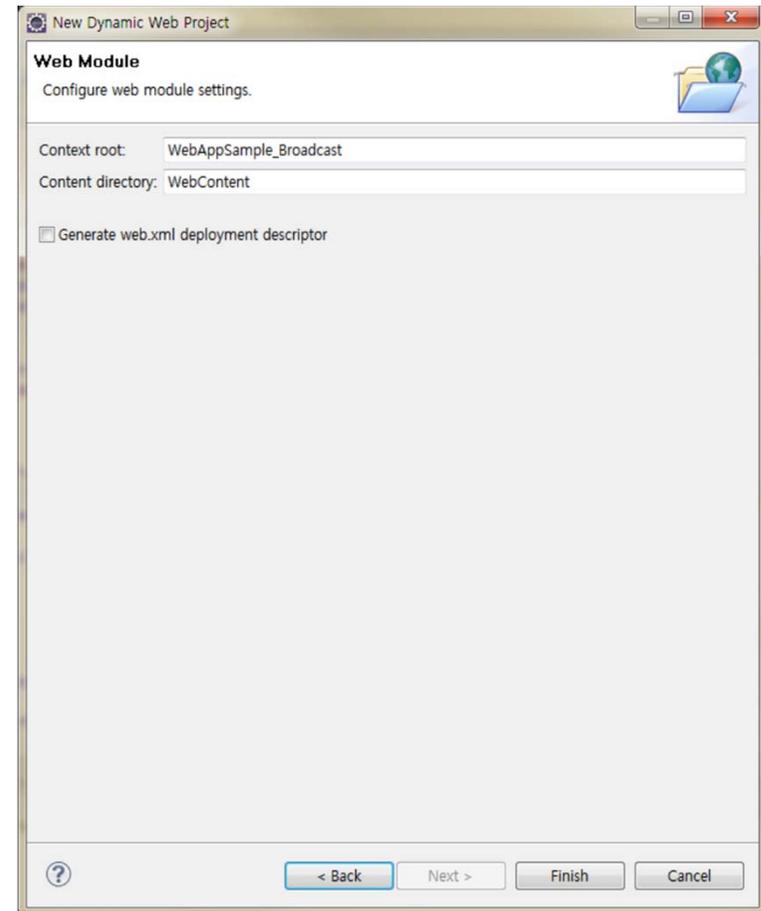
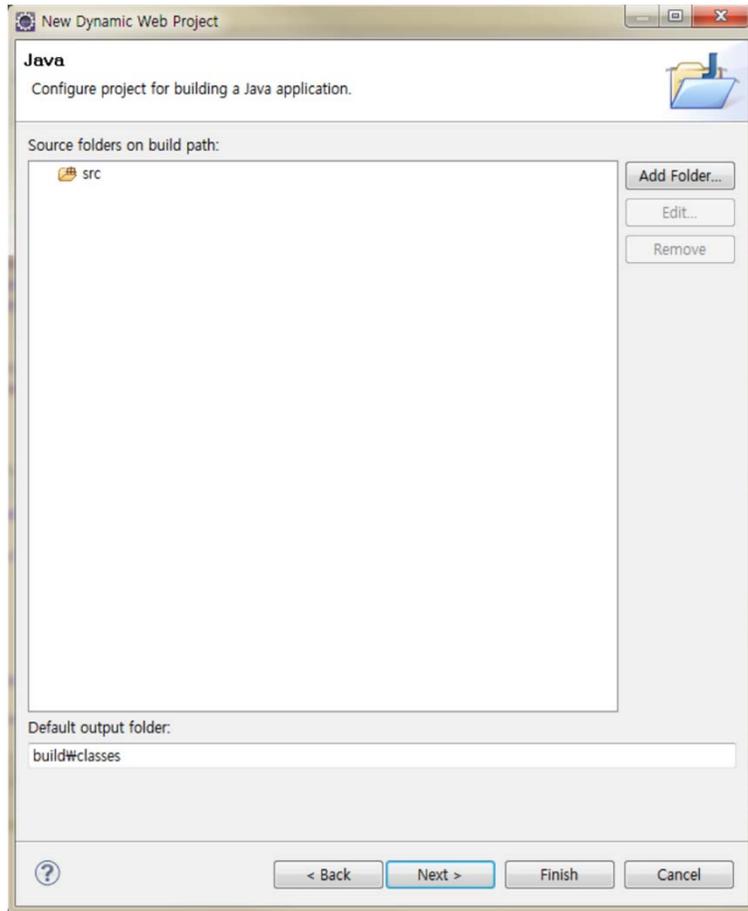


- ✓ Project name 입력
- ✓ wepapp 선택
- ✓ Finish



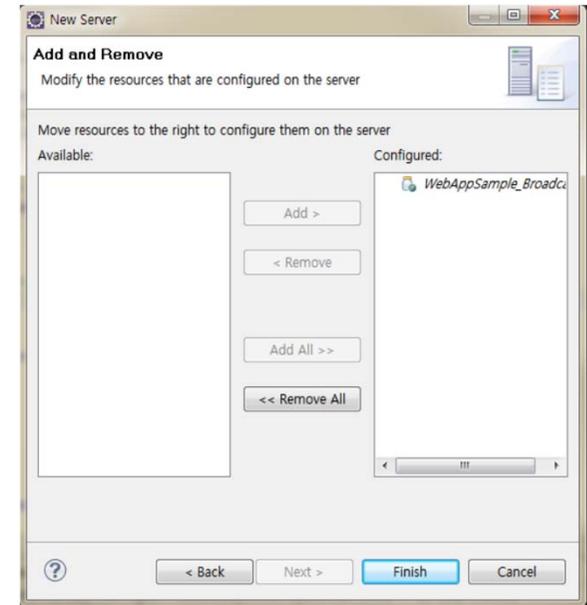
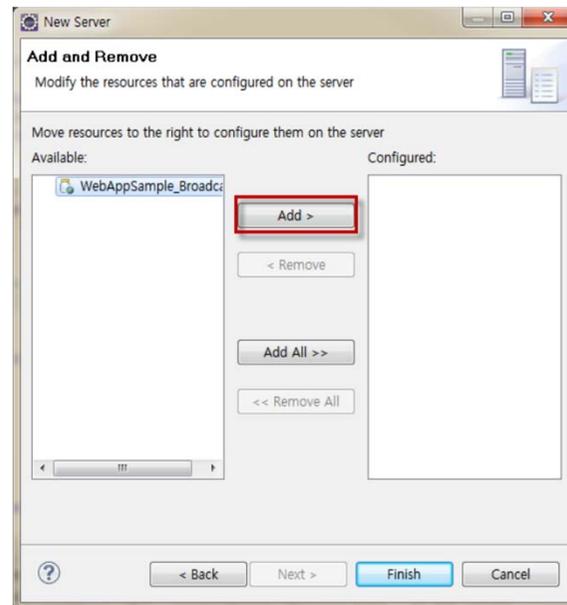
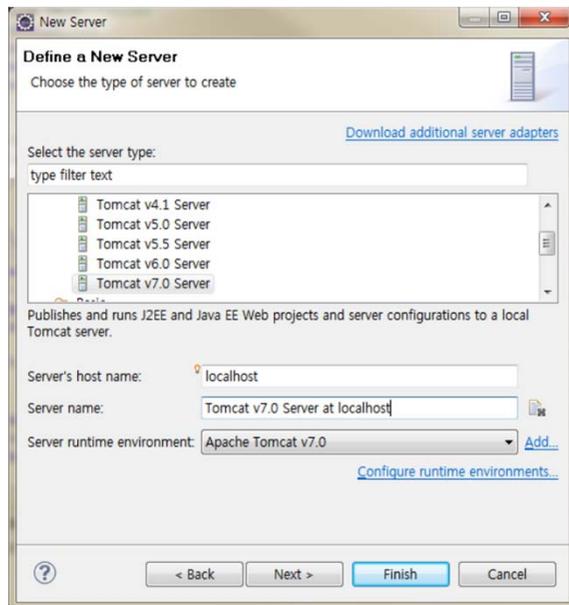
Development Process

■ Creating LG Project



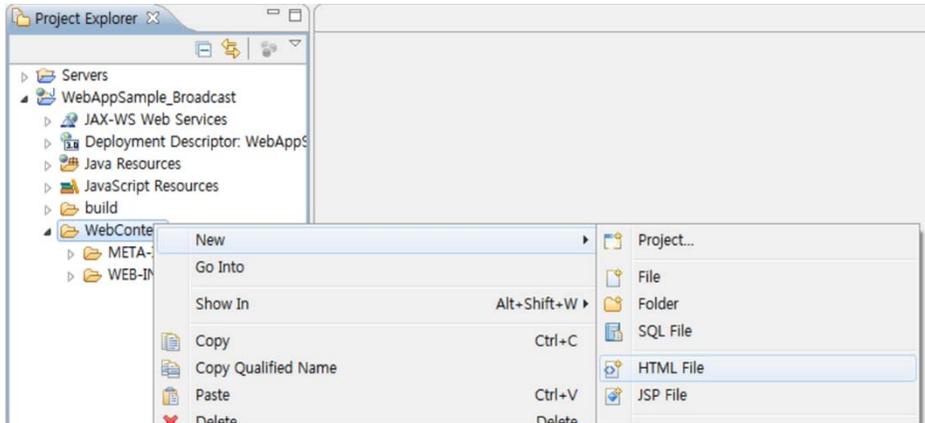
Development Process

■ Creating LG Project

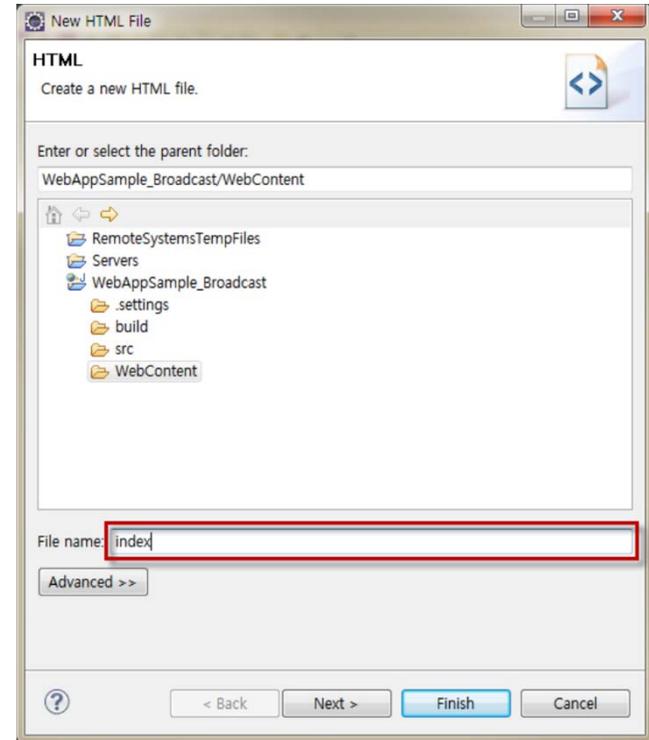


Development Process

■ Creating LG Project



- ✓ HTML 파일 추가
- ✓ JavaScript파일 추가
 - main.jss , keycode.js
- ✓ CSS 파일 추가
 - main.css



Development Process

■ Implementation

✓ JavaScript, CSS 연계하기 (index.html)

- Application 에서 사용할 JavaScript 와 stylesheet 소스 파일의 경로를 지정해 준다

```
<!-- CSS or JavaScript -->  
<link rel="stylesheet" href="webOSjs-0.1.0/CSS/main.css" />  
><script type="text/javascript" src="webOSjs-0.1.0/JS/main.js"></script>  
<script type="text/javascript" src="webOSjs-0.1.0/JS/keynode.js"></script>
```

✓ 키 입력 이벤트(index.html)

- Application 동작 중 발생하는 키 입력을 처리하기 위해 body tag에 on key down event handler를 등록한다.

```
<body onkeydown="processKeyDown(event);">
```

Development Process

■ Implementation

✓ 화면 설계하기 (index.html)

- Application 에서 보여질 화면을 div tag를 사용하여 설계

```
<div class="top_title"></div>
<div class="left_padding"></div>
```

- 방송 화면이 보여질 부분object 타입을 정의하여 사용한다.

```
<div class="broadcast_area">
<!-- Broadcast -->
<object type="application/x-netcast-broadcast"
  id="broadcast"
  width="800"
  height="600">
</object>
```

Development Process

■ Implementation

✓ 화면 설계하기 (index.html)

■ 채널 변경 버튼과 채널 정보를 보여줄 부분

- 채널 변경과 채널 정보를 보여주기 위한 버튼을 만든다. 버튼 클릭 시 발생하는 이벤트를 처리하기 위해 onClick event handler를 정의한다.

```
<div class="channel_control">
<!-- Control Button -->
<button class="btn" type="button" onclick="channelUp()">channel up</button>
<br/>
<button class="btn" type="button" onclick="channelDown()">channel down</button>
<br/>
<button class="btn" type="button" onclick="channelInfo()">Get Channel Info</button>
<br/>
<br/>
```


Development Process

■ Implementation

✓ 화면 설계하기 (index.html)

■ 돌아가기 버튼 및 설계 마무리

- 어플리케이션을 종료하고 App 화면으로 돌아가기 위한 Back 버튼과 현재 방송 채널로 돌아가기 위한 Exit 버튼을 생성한다. 버튼 클릭 시 발생하는 이벤트를 처리하기 위해 onClick Event handler를 정의한다.

```
</div>
<div class="back_exit">
<button class="" type="button" onclick="back()">back</button>
 
<button class="" type="button" onclick="exit()">exit</button>
</div>
</div>

<div class="right_padding">
</div>
<div class="bottom_footer">
</div>

</body>
</html>
```

Development Process

■ Implementation

✓ 동작 설계하기 (main.js)

■ 버튼 클릭 시 동작 이벤트

- 화면의 버튼을 눌렀을 때 LG Web Open API를 사용하여 정보를 읽어오는 함수에 대한 JavaScript code를 구현한다.

현재 방송중인 채널을 올리기 위해 channelUp 메소드를 사용

```
function channelUp(){  
    broadcast.channelUp();  
    delete_channelInfo();  
}
```

Application을 종료하고 my app 화면으로 돌아가기 위해 NetCastBack 메소드를 사용

```
function channelDown(){  
    broadcast.channelDown();  
    delete_channelInfo();  
}
```

현재 방송중인 채널을 올리기 위해 channelDown 메소드를 사용

```
function back(){  
    window.NetCastBack();  
}
```

Application을 종료하고 방송 채널로 돌아가기 위해 NetCastExit 메소드를 사용

```
function exit(){  
    window.NetCastExit();  
}
```

Development Process

■ Implementation

✓ 동작 설계하기 (main.js)

▪ 키 입력 이벤트 처리

- Application 실행 중 발생하는 리모콘 키 입력에 따라 해야 할 동작을 정의한다.

```
function processKeyDown(event)
{
    switch(event.keyCode){
        case VK_PAGE_UP:
            channelUp();
            break;

        case VK_PAGE_DOWN:
            channelDown();
            break;
        case VK_BACK:
            back();
            break;
    }
}
```

Development Process

■ Implementation

✓ 동작 설계하기 (main.js)

■ 채널 정보 얻어오기

- 현재 방송되고 있는 채널의 정보를 얻어옵니다. `getCurrentChannelName` 메소드를 사용하여 채널 이름을 얻어온다. `getCurrentProgram` 메소드를 사용하여 현재 방송되고 있는 프로그램의 EPG 정보를 얻어올 수 있다. 얻어온 정보들을 `span tag`에 표시해 준다.

```
function channelInfo()
{
    var channelName = broadcast.getCurrentChannelName();
    var currentProgram = broadcast.getCurrentProgram();

    var title = currentProgram.title;
    var startTime = currentProgram.startTime;
    var endTime = currentProgram.endTime;
    var description = currentProgram.description;

    document.getElementById("channel_name").innerHTML= channelName;
    document.getElementById("program_title").innerHTML= title;
    document.getElementById("program_starttime").innerHTML= startTime;
    document.getElementById("program_endtime").innerHTML= endTime;
    document.getElementById("program_description").innerHTML= description;
}
```

Development Process

■ Implementation

✓ 동작 설계하기 (main.js)

■ 채널 정보 삭제하기

- 화면에 표시된 채널 정보를 삭제한다. 채널이 변경되는 경우 delete_channelInfo 함수를 호출하여 채널 정보를 삭제한다.

```
function delete_channelInfo(){
    document.getElementById("channel_name").innerHTML = "";
    document.getElementById("program_title").innerHTML = "";
    document.getElementById("program_starttime").innerHTML = "";
    document.getElementById("program_endtime").innerHTML = "";
    document.getElementById("program_Description").innerHTML = "";
}
```

Development Process

■ Result

