

# 임베디드 SW 경진대회: 기술 교육

## Web Application Development



# 목차

- 1. webOS OSE 앱 개발하기
- 2. webOS OSE APIs
- 3. Luna Service API 사용하기
- 4. web socket 통신하기

# webOS OSE에서 앱 개발하기



# 애플리케이션이란?

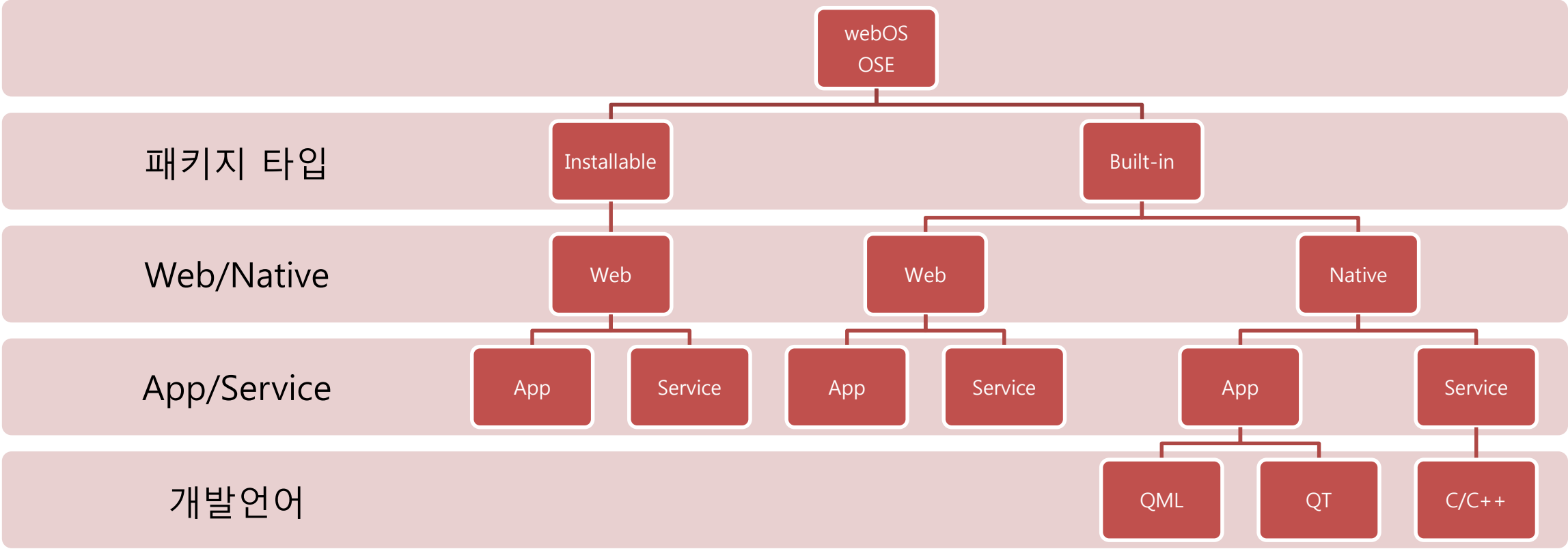
## ● 애플리케이션이란?

- 넓은 의미 : 운영체제에서 실행되는 모든 소프트웨어
  - 워드프로세서, 컴파일러, 링커 등
- 좁은 의미 : 운영체제 위에서 사용자가 직접 사용하는 소프트웨어
  - 워드프로세서, 미디어 플레이어 등
- 설치되는 용도나 형태에 따라
  - 모바일 용 애플리케이션
  - TV용 애플리케이션
  - 웹 애플리케이션
  - 네이티브 애플리케이션
  - 넷플릭스 애플리케이션
  - 등등



[이미지 출처] <https://smnewsnet.com/archives/19077/sheriffs-warning-party-crashers-social-networking/>

# OSE에서 개발 가능한 애플리케이션 타입은?



# webOS에서 사용하는 애플리케이션 패키지

## 🐼 애플리케이션 패키지란?

- 소프트웨어나 애플리케이션, 데이터의 “배포판”
- 애플리케이션을 구성하는 파일들이 포함됨
  - 메타파일, 실행파일, 아이콘, Readme 등

## 🐼 다양한 패키지 타입

- iOS – IPA 패키지
- Android – APK 패키지
- webOS – IPK 패키지
- 우분투 – DEB 패키지, RPM 패키지
- 심비안 – SIS, SISX 패키지

iOS



webOS



# 웹 애플리케이션이란?

---

## 👤 정의

- 인터넷이나 인트라 넷을 통해 웹 브라우저 위에서 이용할 수 있는 응용 소프트웨어
- 웹 표준에 따라 개발 (호환성)

## 👤 구성요소

- 웹 브라우저 : 사용자가 요청한 웹 서버의 결과를 보여주기 위한 애플리케이션
  - 크롬, 파이어폭스, IE 등
- 웹 서버 : 웹 브라우저의 요청에 대한 결과를 응답해주는 기능, 정적인 페이지 처리
  - 아파치, Nginx, IIS 등
- 웹 애플리케이션 서버 : 동적인 페이지 처리
  - 톰캣, 웹로직, 웹 스파이어 등
- 데이터베이스 : 데이터를 저장하는 저장소

# webOS 애플리케이션의 메타파일

<http://webosose.org/develop/configuration-files/appinfo-json/>

## appinfo.json

- webOS 애플리케이션의 메타데이터 파일, 앱 root 디렉토리에 위치
- 애플리케이션이 패키지 되기 위해서 반드시 필요

```
1  {
2      "id": "com.myco.app.appname",
3      "title": "AppName",
4      "main": "index.html",
5      "icon": "AppName_80x80.png",
6      "largeIcon": "AppName_130x130.png",
7      "type": "web",
8      "vendor": "My Company",
9      "version": "1.0.0",
10     "appDescription": "This is an app tagline",
11     "resolution": "1920x1080",
12     "iconColor": "red",
13     "splashBackground": "AppName_Splash.png",
14     "transparent": false,
15     "requiredMemory": 20,
16     "requiredPermissions": ["time", "media"]
17 }
```

### - id: 고유한 App ID.

- ID는 회사 또는 단체의 역도메인으로 시작할 것 (Recommended) ex) com.lge.OOO
- 다음 역도메인으로 시작하지 말 것: com.palm, com.webos, com.lge, com.palmdts
- ID는 서브 도메인인 app.app-name 으로 끝날 것 (Recommended)
- ID는 반드시 영문 소문자(a-z), 숫자 (0-9), 빼기 (minus) 부호, 마침표(.)로만 구성할 것.
- ID는 반드시 alphanumeric 으로 시작해야 하며 최소 2문자(character) 이상으로 구성할 것



# webOS 애플리케이션의 메타파일

<http://webosose.org/develop/configuration-files/packageinfo-json/>

## packageinfo.json

- 서비스 패키지를 위한 메타파일
- CLI로 앱을 패키징 하면 자동으로 생성되는 파일

```
{
  "app": "com.webos.app.testapp",
  "id": "com.webos.testapp",
  "loc_name": "webOS Open Source Edition Demo",
  "package_format_version": 2,
  "vendor": "My Company",
  "version": "1.0.0",
  "services": [
    "com.webos.testapp.service"
  ]
}
```

Property	Required	Type	Description
app	Required	string	Application ID.
id	Required	string	Package ID.
loc_name	Required	string	Localizable package name.
package_format_version	Required	number	Packaging format version, currently, "2".
vendor	Required	string	Vendor name.
version	Required	string	Package version.
services	Optional	string array	Services packaged with the app.

# webOS 애플리케이션의 메타파일

<http://webosose.org/develop/configuration-files/services-json/>

## services.json

- 서비스의 Root 디렉토리에 위치하며, 서비스의 구조를 기술함

```
{
  "id": "com.test.testacct.test.service",
  "description": "Test Service",
  "services": [
    {
      "name": "com.test.testacct.test.service",
      "description": "Test Contact"
    }
  ]
}
```

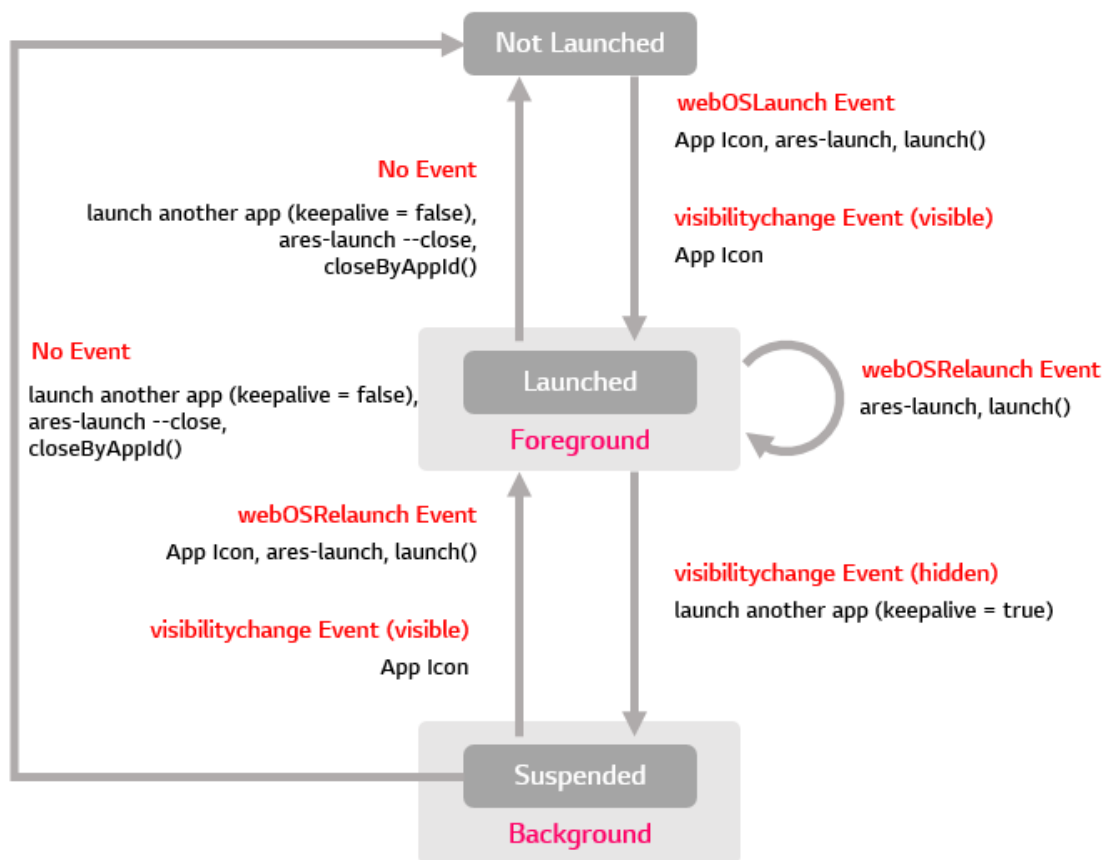
Property	Required	Type	Description
id	Required	string	ID of the service array. Typically, the value will be the name of the first service in services object array.
description	Optional	string	Description of the service array.
services	Required	object array	Services the app provides. Typically, an app provides only one service. However, there may be reasons to provide multiple services within the same app.
-name	Required	string	Name of service on the webOS Luna Bus. The service name must begin with the app name. For example, <ul style="list-style-type: none"><li>• App name: <i>com.example.myapplication</i></li><li>• Service name: <i>com.example.myapplication.myservice</i></li></ul>
-description	Optional	string	Description of the service.

# webOS 애플리케이션의 LifeCycle

<http://webosose.org/develop/web-app-dev/web-app-lifecycle/>

## LifeCycle

- Not Launched / Launched / Suspended 의 세가지 상태를 가짐



```
1 // webOSLaunch event
2 document.addEventListener('webOSLaunch', function(inData) {
3   // Check the received parameters
4   console.log(JSON.stringify(inData.detail));
5
6   // Do something in the foreground
7   ...
8 }, true);
9
10 // webOSRelaunch event
11 document.addEventListener('webOSRelaunch', function(inData) {
12   // Check the received parameters
13   console.log(JSON.stringify(inData.detail));
14
15   // Do something in the foreground
16   ...
17 }, true);
```

```
1 document.addEventListener ('visibilitychange', function() {
2   if (document.hidden)
3     doHiddenCleanup();
4   else
5     doShowingTasks();
6 }, true);
```

# webOS 애플리케이션 로컬라이징

## 🗣 Localization이란?

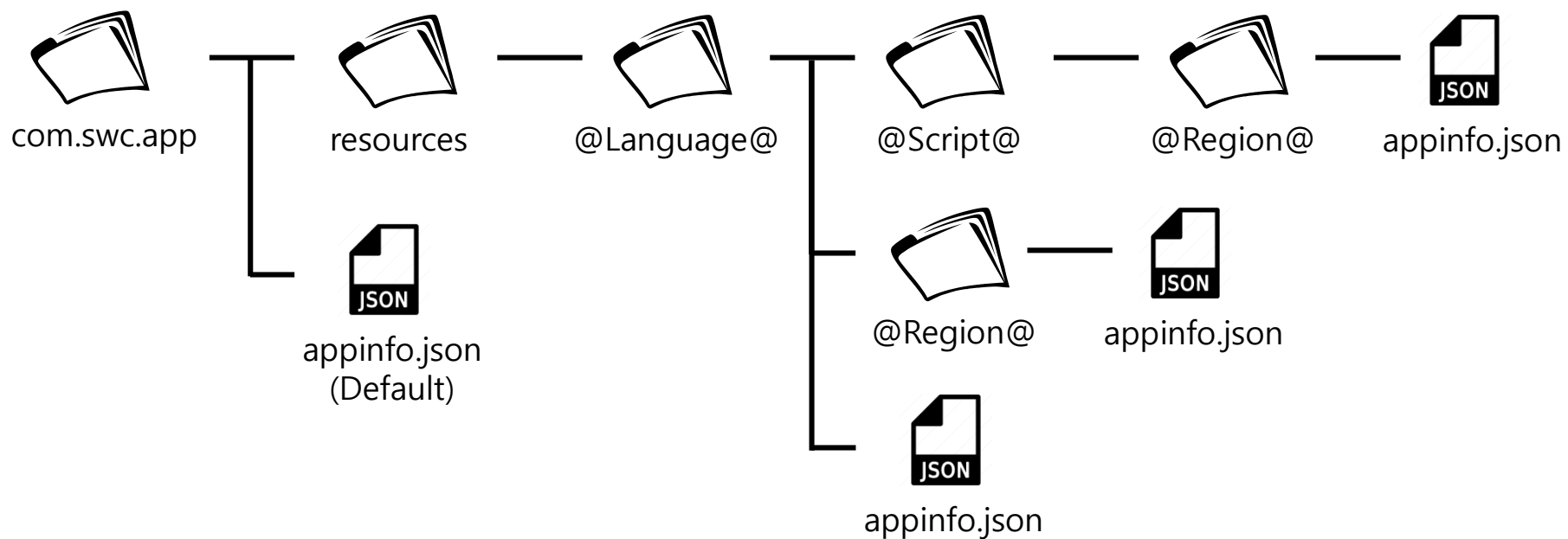
- Localization (현지화)는 “L10N” 이라고 쓰임
- Locale에 따른 리소스를 핸들링 하는 것
- UI/UX를 지역에 맞도록 변경하는 작업

## 🗣 Localization이란?

- Locale은 “Language”, “Script”, “Country/Region”의 조합으로 정의됨 (BCP 47 표준)  
예) en-US, ko-KR, mn-Cyrl, mn-MN, mn-Cryl-MN, mn-Mong-CN
- Wikipedia : <http://en.wikipedia.org/wiki/Locale>
- Language Code : [http://en.wikipedia.org/wiki/List\\_of\\_ISO\\_639-1\\_codes](http://en.wikipedia.org/wiki/List_of_ISO_639-1_codes)
- Region Code : [http://en.wikipedia.org/wiki/ISO\\_3166-1\\_alpha-2](http://en.wikipedia.org/wiki/ISO_3166-1_alpha-2)
- Script Code : [http://en.wikipedia.org/wiki/ISO\\_15924#List\\_of\\_codes](http://en.wikipedia.org/wiki/ISO_15924#List_of_codes),  
[http://ko.wikipedia.org/wiki/ISO\\_15924\\_부호\\_목록](http://ko.wikipedia.org/wiki/ISO_15924_부호_목록)

# webOS 애플리케이션의 메타데이터 로컬라이징

## 📁 appinfo.json 디렉토리 구조



# webOS 애플리케이션의 소스 로컬라이징

## 🔗 JavaScript

- iLib 라이브러리를 사용
- JSON 형태의 strings.json 파일로 구성
- 위치는 appinfo.json 과 동일함

```
{  
  "String 1": "Translation 1",  
  "String 2": "Translation 2"  
}
```

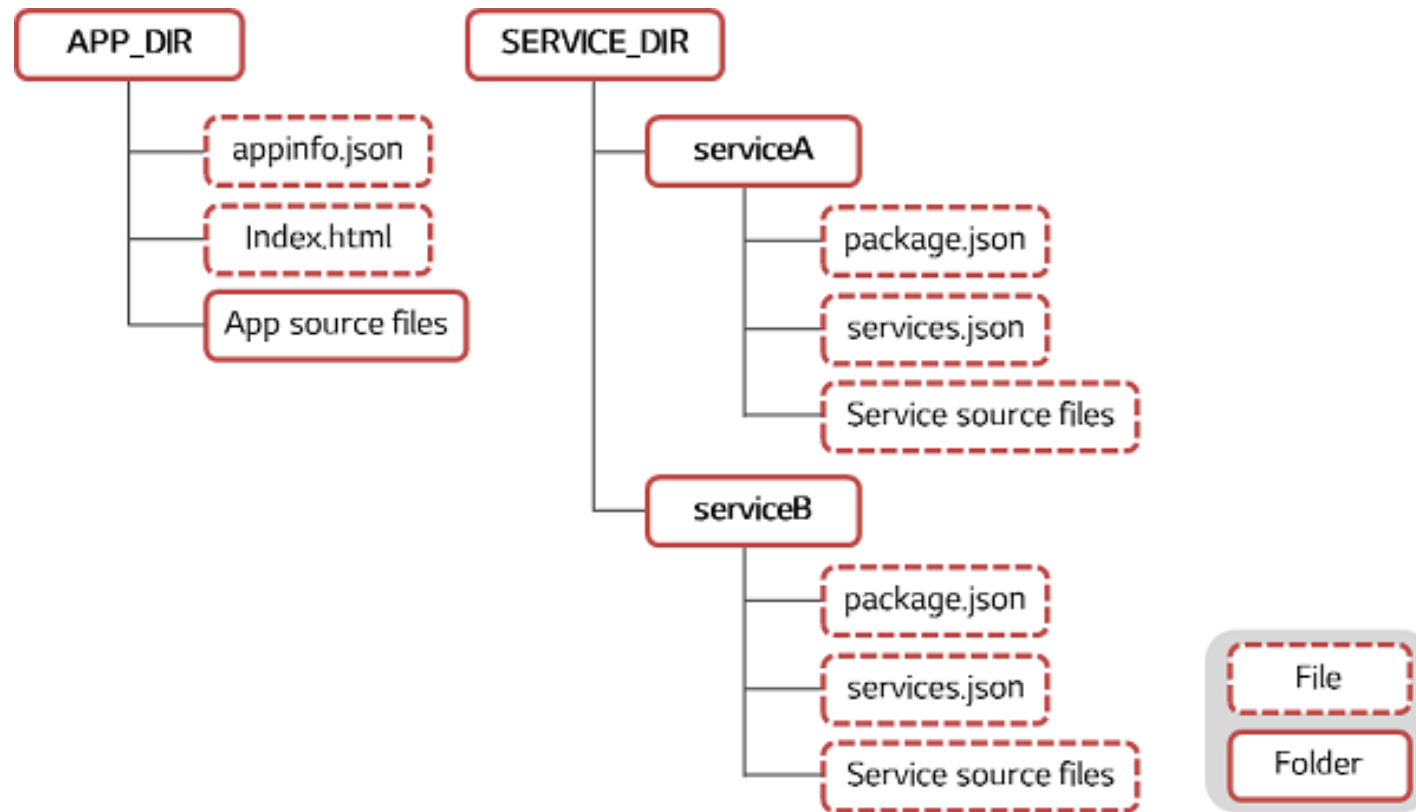
## 🔗 C/C++

- libwebosi18n 라이브러리를 사용
- JavaScript와 같이 JSON 형태의 스트링 리소스를 생성
- 위치는 appinfo.json 과 동일함

# webOS 애플리케이션 / JS 서비스 구조

## 디렉터리 구성

- 서비스 디렉터리는 하위로 여러 개의 서비스를 가질 수 있음



# webOS OSE APIs





# API란?

---

## 👤 애플리케이션 프로그래밍 인터페이스

- API는 Application Programing Interface의 약어
- 운영체제나 프로그래밍 언어가 제공하는 기능을 제어할 수 있게 만든 인터페이스

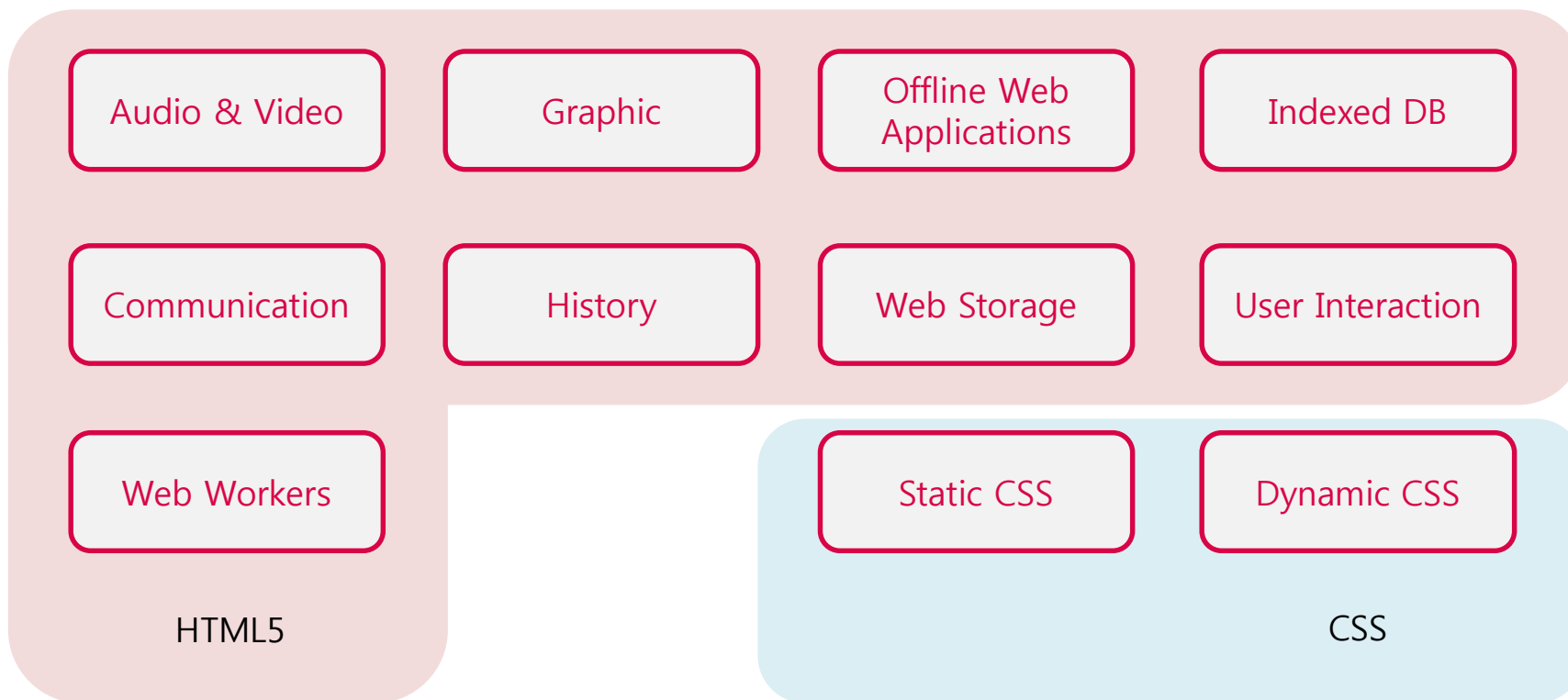
## 👤 예

- 마이크로소프트 API들 : [https://msdn.microsoft.com/library?url=/library/en-us/winprog/winprog/overview\\_of\\_the\\_windows\\_api.asp](https://msdn.microsoft.com/library?url=/library/en-us/winprog/winprog/overview_of_the_windows_api.asp)
- 텐서플로우 : [https://www.tensorflow.org/api\\_docs/](https://www.tensorflow.org/api_docs/)
- 구글 지도 API : <https://developers.google.com/maps/documentation/?hl=ko>
- webOS TV API : <http://webostv.developer.lge.com/api/webos-service-api/>

# Standard Web API?

## 🦉 표준 Web API 들

- W3C 표준 Web API (HTML5, CSS3) 지원



# Luna Service API?



## Luna Service API 들

- webOS OSE 플랫폼의 서비스들이 제공하는 API

Service URI	설명	메소드
com.webos.applInstallService	애플리케이션 설치 요청을 수행하고 설치 중인 애플리케이션 정보를 제공	Install, Remove, status
com.webos.bootManager	부팅 상태/모드를 모니터링하고 제어	generateSignal, getBootStatus
com.webos.media	webOS 미디어 서버에서 제공하는 기능에 대한 액세스를 제공. 앱에서는 이 API를 직접 사용하는 것보다 애플리케이션 프레임워크(web, QT, S이)가 기본 제공하는 미디어 인터페이스를 사용	load, unload, notifyForeground, notifyBackground, subscribe, attach, play, pause, seek, registerPipeline, unregisterPipeline, acquire, tryAcquire, release, trackAppProcesses, contentReady, setVideoInfo, registerMedia, unregisterMedia, focus, switchToFullScreen, setVisibility, unsubscribe, getActivePipelines, setVolume, getForegroundApplInfo
com.webos.notification	시스템 알림 관리	closeAlert, closeToast, createAlert, createToast
com.webos.pmlogd	webOS OSE의 로그 데몬	backuplogs, forcerotate, subscribeOnRotation
com.webos.rdx	충돌, 분석 및 개요 보고서를 생성하는 원격 진단 데몬	getLogSettings, setLogSettings, makeReport, makeLogReport

# Luna Service API?

Service URI	설명	메소드
com.webos.service.activitymanager	시스템의 다양한 부분을 모니터링하고 해당 이벤트가 발생할 때 동작	create, start, complete, cancel, stop, adopt, release, pause, getActivityInfo, getManagerInfo
com.webos.service.ai.voice	음성 입력을 인식	start, getState, getResponse, stop
com.webos.service.alarm	webOS 데몬 또는 앱이 지정된 시간 간격에 특정 매개변수를 사용하여 webOS API를 호출	set, clear
com.webos.service.applicationmanager	애플리케이션 수명 주기, 애플리케이션 정보, LaunchPoint 목록을 관리	closeByAppId, getAppBasePath, getAppInfo, getAppLifeStatus, getForegroundAppInfo, launch, listApps, listLaunchPoints, registerNativeApp, running
com.webos.service.audio	webOS 시스템의 오디오 정책을 제어	개발자 문서 참조
com.webos.service.audiooutput	오디오 소스를 오디오 출력 장치에 연결하고 오디오 장치의 볼륨을 제어. '오디오 출력' 서비스는 HAL 또는 ALSA 또는 기타 하드웨어 추상화 계층에서 작동 (앱에서 직접 사용할 수 없음)	개발자 문서 참조
com.webos.service.avoutput	비디오 소스(디코더 또는 HDMI의 프레임)를 비디오 평면에 연결. 비디오 후 처리 및 스케일링을 위한 인터페이스를 제공 (앱에서 직접 사용하는 것을 권장하지 않음)	개발자 문서 참조

# Luna Service API?

Service URI	설명	메소드
com.webos.service.bluetooth2	webOS 에서 원격 블루투스 장치를 사용	개발자 문서 참조
com.webos.service.config	webOS 구성 요소에 제품별 구성을 제공하는 시스템의 정적 서비스	getConfigs, reconfigure, setConfigs
com.webos.service.configurator	webOS 앱을 위한 데이터베이스 스키마, 파일 캐시 구성, 활동을 생성	scan, rescan, unconfigure
com.webos.service.connectionmanager	webOS 앱과 서비스에서 유/무선 네트워크 연결을 관리하는데 사용	findProxyForURL, getinfo, getStatus, setdns, setipv4, setipv6, setProxy, setstate, setTechnologyState
com.webos.service.db	앱에서 영구적인 데이터를 저장	개발자 문서 참조
com.webos.service.devmode	개발자모드 지원 서비스. 디바이스에서 개발자 모드를 활성화/비활성화하고 개발자 모드 상태를 확인	getDevMode, setDevMode
com.webos.service.downloadmanager	디바이스에서 파일 업로드와 다운로드를 관리	개발자 문서 참조
com.webos.service.filecache	파일 캐시를 관리 (기본캐시 : 100mb)	
com.webos.service.memorymanager	커널 아웃 메모리 상황에 시스템이 실행되지 않도록 애플리케이션을 관리	
com.webos.service.preferences	시스템 및 응용 프로그램 속성을 관리 (기존 시스템 애플리케이션이 이미 사용중인 경우에만 사용해야 함)	

# Luna Service API?

Service URI	설명	메소드
com.webos.service.power	애플리케이션이 시스템 일시 중단/재개 주기와 상호작용할 수 있음	
com.webos.service.rosbridge	webOS 애플리케이션이 ROS2 디바이스의 서비스와 통신하기 위한 서비스	publish, subscribe, call
com.webos.service.settings	webOS에서 사용되는 설정을 관리	
com.webos.service.sleep	애플리케이션이 시스템 일시 중단/재개 주기와 상호작용할 수 있음	
com.webos.service.systemservice	시스템에 대한 액세스를 제공 (backup option, Getting Device Info, Getting and modifying the date and time settings, Setting ringtones, Wallpaper, Local Formats)	
com.webos.service.tempdb	DB8 서비스와 같음	
com.webos.service.webappmanager	WAM : 웹 기반 애플리케이션을 처리하기 위한 내부 시스템 서비스	
com.webos.service.wifi	Wifi 연결을 관리하기 위한 서비스	

# web application 에서 Luna Service API 사용하기



# Luna Service API 사용하기

---

## Luna Service API의 구성

- Service URI
  - 서비스의 URI (예: luna://com.webos.service.devmode 형태)
- Method name
  - 서비스의 메소드 이름
- Asynchronous function call
  - Success/Failure Callback
  - Complete Callback
- Request Object
- Response Object



# Luna Service API 사용하기

## 🐧 webOS.js 라이브러리 추가하기

- `<script type="text/javascript" src="webOSjs-0.1.0/webOS.js"></script>`

## 🐧 webOS.service.request() 함수 구현

```
var request = webOS.service.request("luna://com.mycom.helloworld.service", {
  method: "hello",
  parameters: {"input": inputMessage},

  onSuccess: function (args) {
    //TO-DO
  },

  onFailure: function(inError) {
    //TO-DO
  },

  onComplete: function(inResponse) {
    //....
  },

  subscribe: subscribeStatus ,
  resubscribe: resubscribeStatus
});
```

# Luna Service API 사용하기

## 🔔 appinfo.json에 ACG 설정

- Assume there is method123 at this URI: com.webos.service.requiredservice/method123
1. Get ACG (Access Control Group) permission for the method
    - You must update the **appinfo.json** file to include the **requiredPermissions** parameter and specify the ACG group(s) of the required methods.

appinfo.json - Add 'group1' and 'group2' so that the app has access to 'com.webos.service.requiredservice/method123'

```
{  
  "id" : "com.webos.app" ,  
  ...  
  "requiredPermissions": ["group1", "group2"],  
  ...  
}
```

No "requiredPermissions" parameter on default

# web application 에서 Luna Service API 사용하기 - 실습



# Luna Service API 사용하기

---

## 🐼 `com.webos.service.systemservice` 를 사용해 Locale 값 구하기

- Service URI : `com.webos.service.systemservice`
- method : `getPreferences`
- Parameters : `{"keys" : ["locale"]}`

# Web Application 개발 실습

---

## 해보기

- 템플릿 목록 보기 : `ares-generate -l`
- 빈 프로젝트 생성 : `ares-generate -t basic first_webapp`
  - App ID : `com.knu.app`
  - Title : KNU sample app
  - Version : 1.0.0
- 소스 수정
  - Service URI : `com.webos.service.connectionmanager`
  - method : `getStatus`
  - Output : wifi 와 wired IP 주소 구하기
- 패키지 생성 : `ares-package first_webapp`
- 에뮬레이터에 설치 : `ares-install -d <에뮬레이터 이름> <패키지 파일명>`
- 디바이스에 설치 : `ares-install -d <디바이스 이름> <패키지 파일명>`

# Web Application 개발 실습

## 해보기

```
<script type="text/javascript">
//sample code for calling LS2 API
var lunaReq= webOS.service.request("luna://com.webos.service.connectionmanager",
{
    method:"getStatus",
    parameters: {},
    onSuccess: function (args) {
        console.log (args.wired);
        document.getElementById("title").innerHTML = "wired : " + args.wired.ipAddress + " , wifi : " + args.wifi.ipAddress ;
        webOS.debug("wired : " + args.wired.ipAddress + " , wifi : " + args.wifi.ipAddress);
    },
    onFailure: function (args) {
        document.getElementById("title").innerHTML = args.returnValue;
        if (args.returnValue == false){
            webOS.debug(args.errorCode + " : " + args.errorText);
        }
    }
});
</script>
```

# web application 에서 web socket 통신하기 - 실습



# WebSocket 통신 사용하기



<http://webosose.org/blog/websocket-communication-on-webos/>

## WebSocket Communication on webOS

Version added 09-Apr-2018 | Modified 15-Apr-2019

Author: Jaeduck Oh

There are many ways that web apps can use to communicate with the system or other apps. Among those, this article describes how to set up communication using WebSocket.

Check the Wikipedia page for the WebSocket protocol at <https://en.wikipedia.org/wiki/WebSocket>.

### Creating WebSocket Server

The first step is to create a WebSocket server.

You will need:

- Host PC: Any PC with Python works fine, but this article is based on Linux.
- Python 3.x (This article is tested on Python 3.5.3)

1. Download and install a server module using PIP.



You can find an example of a simple WebSocket server at <https://github.com/dpallot/simple-websocket-server>.

```
yourLinux/websocket_server_python$ pip install SimpleWebSocketServer
```

#### Contents

- Creating WebSocket Server
- Creating WebSocket Client
- Writing Result Using Luna Service Notification API
- Installing and Launching App on webOS
  - Packaging App
  - Device Registration
  - Installing App
  - Launching App



# JS Service 개발



# JS Service 개발 실습

## Sample JS Service

- ares-generate의 js\_service template

```
nicolas@nicolas-14U530-MFLGL:~/kbwork$ ares-generate -t js_service first_webservice
? service id com.knu.app.service
Generating js_service in /home/nicolas/kbwork/first_webservice
Success
nicolas@nicolas-14U530-MFLGL:~/kbwork$ ls first_webservice/
helloclient.js  helloworld_webos_service.js  package.json  services.json
nicolas@nicolas-14U530-MFLGL:~/kbwork$
```

File	Description
helloclient.js	Sample JS service which subscribes helloworld_webos_service.js service. This sample shows how to communicate between services.
helloworld_webos_service.js	Sample JS service which provides several simple commands. These commands are specified in the services.json file.
package.json	Configuration file of NPM. For details, see <a href="https://docs.npmjs.com/getting-started/using-a-package.json">https://docs.npmjs.com/getting-started/using-a-package.json</a> .
services.json	Configuration file that defines what commands the service provides on the webOS bus.

# JS Service 개발 실습

## 🔗 해보기

- 템플릿 목록 보기 : `ares-generate -l`
- 빈 프로젝트 생성 : `ares-generate -t js_service first_webservice`
  - Service ID : `com.knu.app.service`
- 패키지 생성 : `ares-package first_webapp first_webservice`
- 에뮬레이터에 설치 : `ares-install -d <에뮬레이터 이름> <패키지 파일명>`
- 디바이스에 설치 : `ares-install -d <디바이스 이름> <패키지 파일명>`
- 디바이스에서 확인 :

```
nicolas@nicolas-14U530-MFLGL:~/kbwork/first_webapp$ ares-shell -d rpi -r "luna-send -n 1 -f luna://com.knu.app.service/hello '{}'"
{
  "message": "Hello, World!",
  "returnValue": true
}
```

# JS Service 개발 실습

---

## 해보기

- github 에서 샘플 가져오기 : [https://github.com/nickyzero/websocket\\_service](https://github.com/nickyzero/websocket_service)
- 패키징/설치 후 실행해보기
- 빈바이저로 모니터링 해보기
- 더 해보기
  - 웹 앱에서 버튼을 누르면 서비스가 Notification 을 띄우는 메소드 만들어보기
  - 웹 앱에서 버튼을 누르면 서비스가 web Socket 서버로 메시지를 보내고 서비스가 리턴 메시지를 받아서 웹 앱에 Notification으로 표시하기

# Arduino 개발



# Arduino WebSocket Server 개발 실습

---

## 해보기

- 아두이노 개발 환경 설정
  - PC에 드라이버 설치
  - 아두이노 IDE 설치
  - 빌트인 LED 제어 테스트
  - <https://deneb21.tistory.com/591>
- ESP8266 WebSocket Server 구현하기
  - websocket library 설치
  - websocket 구현
  - <https://tttapa.github.io/ESP8266/Chap14%20-%20WebSocket.html>
  - [http://www.hardcopyworld.com/gnuboard5/bbs/board.php?bo\\_table=lecture\\_esp&wr\\_id=44](http://www.hardcopyworld.com/gnuboard5/bbs/board.php?bo_table=lecture_esp&wr_id=44)
  - [ 참조코드 ]  
[https://github.com/godstale/ESP8266\\_Arduino\\_IDE\\_Example/blob/master/example/WebSocketServer/WebSocketServer.ino](https://github.com/godstale/ESP8266_Arduino_IDE_Example/blob/master/example/WebSocketServer/WebSocketServer.ino)

# Enact 개발환경



# Enact 개발환경

## 🐙 Node.js & npm 설치

Node version : 6.5.0  
Npm version : 3.10.3

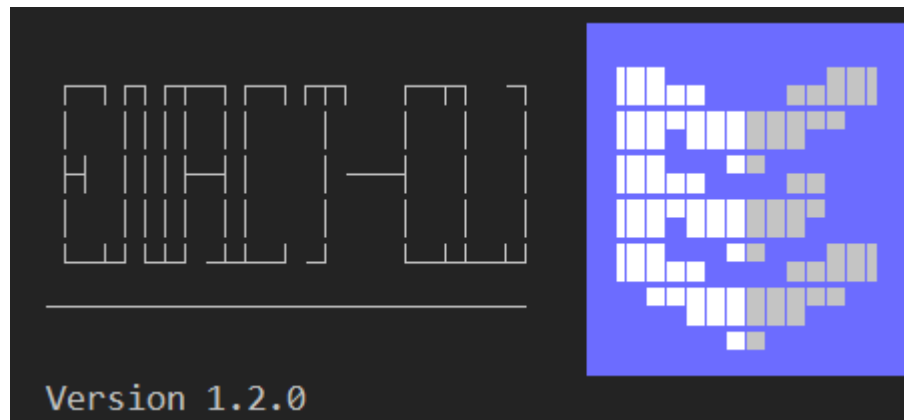
- Nvm 설치 (curl 또는 wget 중 하나)
  - `$ curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.33.11/install.sh | bash`
  - `$ wget -qO- https://raw.githubusercontent.com/creationix/nvm/v0.33.11/install.sh | bash`
  - 설치확인 : `$ command -v nvm`
- Node.js & npm 설치
  - `$ nvm install v6.5.0`
  - npm은 자동으로 3.10.3 버전으로 설치됨
  - Node 버전 확인 : `$ node --version`
  - npm 버전 확인 : `$ npm --version`



# Enact 개발환경

## 🐧 Enact 설치

- npm 명령어로 설치 (root 권한으로 설치)
  - `$ npm install -g @enact/cli`
  - 설치확인 : `$ enact -v`
- 템플릿 생성
  - `$ enact create helloworld`
  - `$ cd helloworld`
  - `$ npm install`
  - `$ npm run serve`
- 브라우저에서 확인



# ABOUT webOS ?

QUESTIONS ?

? ANSWERS



# webOS

Open Source Edition

