

# Morpheus<sup>U</sup>

Client Guide

교육자료



# Client Guide 주요 목차

---

## 1. IDE 설치

## 2. Client IDE 소개

1. Client IDE 개요
2. M-SDK( Morpheus 라이브러리 매니저)
3. 프로젝트 생성
4. Manifest.xml

## 3. M.interface 소개

1. MCore?
2. M.interface 메서드
3. 모피어스 공식 Wiki

## 4. Javascript Style Guide

## 5. 화면 개발

1. 개발 주의사항

## 6. 배포

## 7. Q&A

# IDE 설치

---

## 1. JDK (Java Development Kit) 다운로드

<http://www.oracle.com/technetwork/java/javase/downloads/index.html>

## 2. Android SDK 다운로드

<http://developer.android.com/intl/ko/sdk/installing/index.html>

## 3. IDE 다운로드

\* WiFi 정보: Uracle\_2F/u34794400  
Uracle\_3F/u34794400

# Client IDE 소개

## #Client IDE 개요

### 1. 모피어스 IDE(Integrated Development Environment)개요

#### 1.1. 구성

1. IDE는 Eclipse Plug-in 기반으로 개발되었으며, Eclipse와 동일한 형태의 개발 환경을 제공합니다.

#### 1.2. 실행환경

1. 권장 OS: Window XP 이상, Mac OS X 10.7(64bit) 이상  
버전에 맞는 실행 파일을 다운로드 받아 설치해야 합니다.

#### 1.3. IDE 문의

- 1.3.1. [www.morpheus.co.kr](http://www.morpheus.co.kr) 기술지원 게시판

# Client IDE 소개

## #M-SDK

M-SDK Manager

Packages

이름	API	버전	상태
▼에뮬레이터			
● 브라우저 에뮬레이터		1.1.9.6	업데이트 가능 - 새 버전 : 1.1.9.7
▼라이브러리			
▼2.1	2.1		
● 코어 라이브러리	2.1	5.1	업데이트 가능 - 새 버전 : 5.3
▼Addon 라이브러리			
● 애드온 Media 라이브러리	2.1	4.7	설치 됨
● 애드온 DB 라이브러리	2.1	4.4	업데이트 가능 - 새 버전 : 4.5
● 애드온 팝업 라이브러리	2.1	4.10	설치 됨
● 애드온 File 라이브러리	2.1	4.5	업데이트 가능 - 새 버전 : 4.6
● 애드온 네트워크 라이브러리	2.1	4.3	업데이트 가능 - 새 버전 : 4.4
● 애드온 Zip 라이브러리	2.1	4.2	설치 됨
● 애드온 네트워크 확장 라이브러리	2.1	4.7	업데이트 가능 - 새 버전 : 4.8
● 애드온 Locale 라이브러리	2.1	0.1	설치 됨
▼Plugin 라이브러리			
● 플러그인 Location 라이브러리	2.1	0.2	설치 됨
● 플러그인 QRCode Reader 라이브러리	2.1	4.5	설치 됨
● 플러그인 Viewer 라이브러리	2.1	1.1	업데이트 가능 - 새 버전 : 1.2
● 플러그인 Drawing 라이브러리	2.1	1.2	설치 됨
● 플러그인 ... 라이브러리	2.1	1.1	업데이트 가능 - 새 버전 : 1.2

목록 업데이트

리소스 설치

리소스 삭제

리소스 가져오기

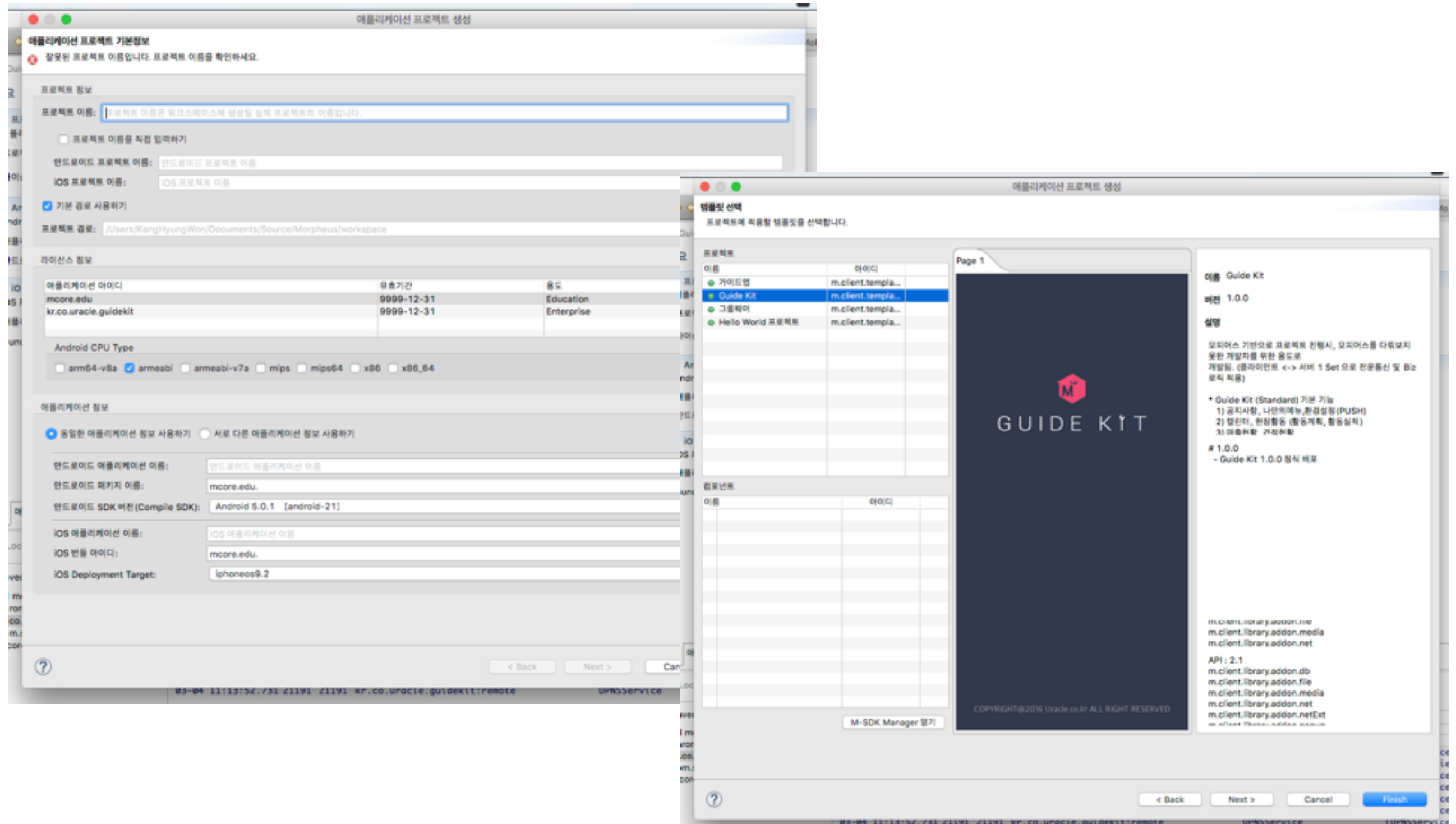
리소스 내보내기

Cancel

OK

# Client IDE 소개

## #프로젝트 생성



# Client IDE 소개

## #Manifest.xml

### 개요

#### 프로젝트 기본 정보

애플리케이션 프로젝트에 대한 기본 정보들입니다.

프로젝트 이름:

라이선스 상태: **Applied**

[변경하기](#)

#### Android 프로젝트 기본 정보

Android 프로젝트에 대한 기본 정보들입니다.

애플리케이션 이름:

안드로이드 패키지 이름:

#### iOS 프로젝트 기본 정보

iOS 프로젝트에 대한 기본 정보들입니다.

애플리케이션 이름:

Bundle Identifier:

#### 설정 페이지들

프로젝트에 관한 설정 페이지들입니다. 모두 두 개의 페이지로 구성되어 있습니다.

[매니페스트](#) : 매니페스트 설정.

[라이브러리 매니저](#) : 라이브러리 관리/적용.

#### 실행 / 배포

실행과 배포에 관련된 바로가기들입니다.

실행하기     **다이렉트 뷰: 사용안함**

[안드로이드 디바이스/에뮬레이터에서 애플리케이션 실행하기](#)

[아이폰 디바이스/시뮬레이터에서 애플리케이션 실행하기](#)

[브라우저 에뮬레이터에서 애플리케이션 실행하기](#)

배포하기

[안드로이드 패키지로 \(APK\) 배포하기](#)

[iOS 패키지로 \(IPA\) 배포하기](#)

[애플리케이션 프로젝트를 압축파일 \(Zip\)으로 배포하기](#)

[애플리케이션 프로젝트 리소스를 압축파일 \(Zip\)로 배포하기](#)

프로젝트들

[iOS 프로젝트를 Xcode 에서 열기](#)

개요   **매니페스트**   라이브러리 매니저   Manifest.xml

# Client IDE 소개

## #Manifest.xml

① 매니페스트 Manifest 파일의 기본적인 세팅 정보들입니다. 각 항목을 선택하시면 해당 항목에 대한 설명을 확인할 수 있습니다.

기본 설정

애플리케이션의 기본 설정들입니다.

Log:

☒ 사용 ☐ 사용안함

Language:

ko

Base Version:

100001

Target:

app

Start Page Name:

www/html/maps.html

Start Page Orientation:

default

Start Page Orientation(Tablet):

default

Direct View:

☐ 사용 ☒ 사용안함

Default Orientation:

default

Default Orientation(Tablet):

default

Default Animation:

default

네트워크 설정

리소스 업데이트

애플리케이션

설명

### 개요

1. 어플리케이션 기본 환경 설정을 하기 위한 Manifest.xml을 제공한다.
2. 플랫폼에서는 어플리케이션 실행시 자동적으로 Manifest.xml의 정보를 읽어 사용하게 된다.
3. Manifest.xml의 위치 : [리소스\(res\) 폴더 구조](#) 참조

### 앱 및 리소스 배포시 주의 사항

1. Manifest.xml은 앱내부(asset(Android) 이나 bundle(iOS))의 파일이 사용되므로, 리소스 배포 파일에 포함되지 않도록 주의 한다.
2. <resource><target> 을 'doc'로 설정한다.
3. <resource><update><mode> 를 'real'로 설정한다.
4. <log> 를 'n'으로 설정한다.

개요 매니페스트 라이브러리 매니저 Manifest.xml



# Client IDE 소개

## #Manifest.xml

### 라이브러리 매니저

#### ▼ 코어 라이브러리 API

프로젝트에서 사용중인 코어 라이브러리 API를 변경합니다.

이름	API	버전	상태
● 코어 라이브러리	2.1	5.1	사용 중

API 변경하기

새로 적용하기

새 버전으로 업데이트하기

#### ▼ 애드온/플러그인 라이브러리

프로젝트에서 사용중인 애드온/플러그인 라이브러리를 변경합니다.

이름	API	버전	상태
▼ Addon 라이브러리			
● 애드온 DB 라이브러리	2.1	4.4	사용 중
● 애드온 File 라이브러리	2.1	4.5	사용 중
● 애드온 Locale 라이브러리	2.1	0.1	미사용
● 애드온 Media 라이브러리	2.1	4.7	사용 중
● 애드온 네트워크 라이브러리	2.1	4.3	사용 중
● 애드온 네트워크 확장 라이브러리	2.1	4.7	사용 중
● 애드온 팝업 라이브러리	2.1	4.10	사용 중
● 애드온 Zip 라이브러리	2.1	4.2	사용 중
▼ Plugin 라이브러리			
● 플러그인 Drawing 라이브러리	2.1	1.2	미사용
● 플러그인 Location 라이브러리	2.1	0.2	미사용

새로 고침

라이브러리 적용

라이브러리 적용 해제

새 버전으로 업데이트하기

개요   매니페스트   라이브러리 매니저   Manifest.xml

#### 설명

이름 코어 라이브러리

API 2.1

버전 5.1

상태 사용 중

#### 설명

하이브리드 모바일 플랫폼 서비스를 제공하기 위한 핵심 엔진부분입니다.

#### 히스토리

# 2.1.5.1 ( 2016.02.05 )

- [Android, iOS] 라이선스 로직 일부 수정
- [iOS] wnInterface 2.2 api 일부 추가
- [Android] HTTP 307 코드에 대한 리다이렉트 핸들러 적용 enable 가능하도록 변경

# 2.1.4.19 ( 2016.01.28 )

- [Android] 불필요한 로그 제거 및 코어 리소스 추가

# 2.1.4.18 ( 2016.01.22 )

- [Android] 유효하지 않은 SSL 페이지 허용을 위한 메소드 추가

# Client IDE 소개

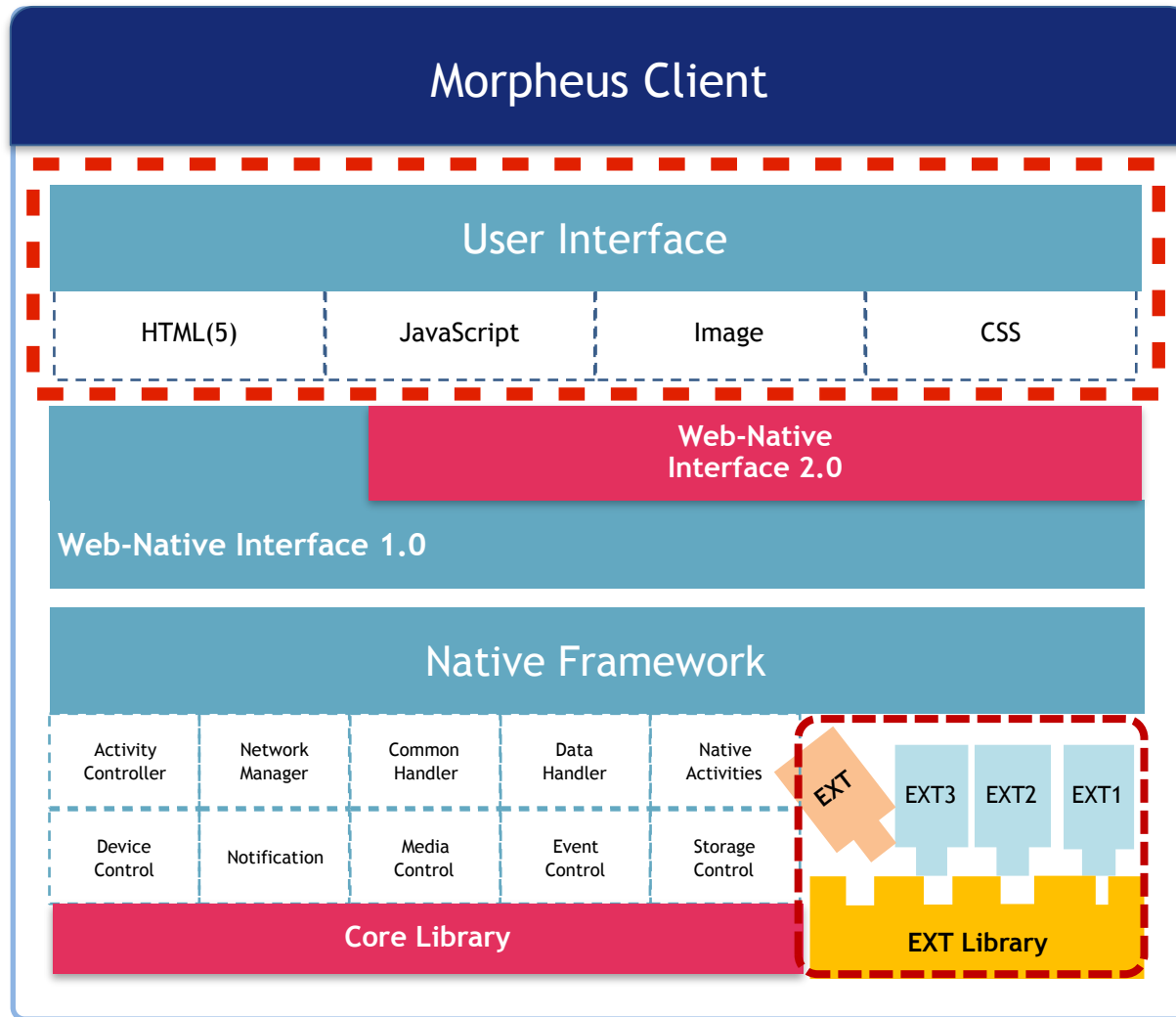
## #Manifest.xml

```
1 <?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
2 <manifest useDirectView="false">
3     <log>y</log>
4     <lang>ko</lang>
5     <resource>
6         <base_version>100001</base_version>
7         <target>app</target>
8         <update>
9             <server/>
10            <trcode/>
11            <mode/>
12        </update>
13    </resource>
14    <startpage>
15        <name>www/html/maps.html</name>
16        <orient>default</orient>
17        <orient-tablet>default</orient-tablet>
18    </startpage>
19    <settings/>
20    <default_action>
21        <orient>default</orient>
22        <orient-tablet>default</orient-tablet>
23        <animate>default</animate>
24        <indicator>n</indicator>
25    </default_action>
26    <library>
27        <ext>mcore.edu.sample.implementation.ExtendWNInterface</ext>
28        <addons>
29            <path>m.client.library.addon.db.WNInterfaceDB</path>
30            <path>m.client.library.addon.file.WNInterfaceFile</path>
31            <path>m.client.library.addon.media.WNInterfaceMedia</path>
32            <path>m.client.library.addon.net.WNInterfaceNet</path>
33            <path>m.client.library.addon.netext.WNInterfaceNetExt</path>
34            <path>m.client.library.addon.popup.WNInterfacePopup</path>
```

개요 매니페스트 라이브러리 매니저 Manifest.xml

# M.interface 소개

## #MCore?



구분	구분	내용
Morpheus Client	Web Browser	단말 내장 브라우저
	Web kit Engine	Web Browser상의 부가적 기능을 추가제공
	Activity Controller	서비스 동작 상태 관리
	Networking Manager	통신 전문 관리 및 TCP/IP 통신관리
	Common Handler	공통 기능 모듈 관리
	Native Activities	Native 서비스 동작 상태 관리
	Event Control	서비스 오류, 실행 내역 관리
	Data Handler	데이터의 변경, 삭제 등의 처리
	Storage Control	저장공간 관리
	EXT Library	추가 확장 기능 개발이 가능한 Plug-In 라이브러리

# M.interface 소개

## #M.interface 메서드

### 1. 상태 이벤트 메서드

M.onReady, M.onRestore, M.onBack, etc..

### 2. 화면 이동

M.page.html, M.page.back, M.page.native, etc..

### 3. 네트워크 통신

M.net.http.send, etc..

### 4. 팝업

M.pop.alert, M.pop.date, M.pop.list, M.pop.instance

### 5. 미디어

M.media.camera, M.media.library, etc..

### 6. 파일

M.file.list, M.file.copy, etc..

### 7. 기타

# M.interface 소개

## #Morpheus 공식 위키

<http://docs.morpheus.kr/mcore/main.php>

The screenshot shows the Morpheus API documentation website. The left sidebar contains a search bar and a navigation menu with categories: ABOUT MORPHEUS (Overview), CLIENT, API (expanded), Core, Addon, Plugin, IDE, UIKit, and SERVER. The main content area is titled 'API' and lists two main categories: Core and Addon. The Core category includes: Common Event, Event Flow, Navigator Object, Page API, Data API, Security API, Info API, Apps API, System API, and Tool API. The Addon category includes: MDatabase (Db API), MFile (File API), MMedia (Media API), MNetwork (Net API), MNetwork Extends (NetExt API), MPopup (Popup API), and MZip (Zip API). A 'View page source' link is visible in the top right corner.

Morpheus

Search docs

ABOUT MORPHEUS

Overview

CLIENT

API

Core

Addon

Plugin

IDE

UIKit

SERVER

Docs » API

View page source

### API

- Core
  - MCore
    - Common Event
    - Event Flow
    - Navigator Object
    - Page API
    - Data API
    - Security API
    - Info API
    - Apps API
    - System API
    - Tool API
- Addon
  - MDatabase
    - Db API
  - MFile
    - File API
  - MMedia
    - Media API
  - MNetwork
    - Net API
  - MNetwork Extends
    - NetExt API
  - MPopup
    - Popup API
  - MZip
    - Zip API

# Javascript Style Guide

## # 할당 ( Assignment )

- 할당 후에는 항상 세미콜론을 넣도록 한다.
- 선언하고 나서 바로 할당할때는 항상 같은 줄에서 하도록한다.
- 선언과 할당이 동시에 하는 변수는 뒤에 기술한다.

```
var foo = false;  
var bar = true;  
var a;  
var b;  
var c;
```

```
var a, b, c,  
    foo = false,  
    bar = true;
```

# Javascript Style Guide

## # 네이밍 ( Naming )

- 이해하기 쉬운 단어를 조합하여 작성하며 축약어나 함축어를 최소화 한다.
- 변수명 앞에 데이터 타입이 중요할 때에는 데이터 타입을 유추 할 수 있는
- 단서를 넣어주는 것도 좋다. (헝가리안 표기법)
- 변수명은 낙타표기법(CamelCase)을 사용하며 상수값의 경우는 언더바를 사용한다.

```
var gU = 'foo';  
var gu = 'foo';  
var groupUnit = 'foo';  
var group_unit = 'foo';  
  
var strName = 'Dragon';  
var arrName = ['Dragon', 'Sword'];  
  
var unitProbe = 'probe';  
var CONST_UNIT_ANGEL = 1004;
```

# Javascript Style Guide

## # 세미콜론 ( Semicolon )

- 자바스크립트에서 기본적으로는 어떠한 두 토큰 사이에서도 줄바꿈으로 구분할 수 있게
- 해석하지만 코드 난독화 작업등의 이유로 소스가 한줄로 합쳐지면서 오류가 발생할 수 있기 때문에 세미콜론의 구분은 중요하다.
  - 세미콜론을 반드시 써야하는 경우와 세미콜론을 꼭 쓰지 않아도 되는 경우는 다음과 같다.

```
var a = 1; // semicolon here

var foo = function() {
    return true; // semicolon here
}; // semicolon here

var factory = function() {
    return {
        }; // semicolon here
}; // semicolon here

function foo() {
    return true;
} // no semicolon here
```



# Javascript Style Guide

## # 띄어쓰기 ( Spacing )

- 탭으로 들여쓰기를 한다.
- 줄 끝 공백은 허용하지 않는다.
- 빈 줄에 공백은 허용하지 않는다.
- 코드내에서는 자유롭게 띄어쓰기 한다.
- 비어있는 객체와 배열에는 괄호 사이에 공백을 넣지 않는다.

```
if(condition) doSomething();  
  
while(condition) iterating++;  
  
for(var i=0;i<100;i++) someIterativeFn();
```

```
if ( condition ) {  
    // Code  
} else if ( condition ) {  
    // Code  
} else {  
    // Code  
}  
  
while ( condition ) {  
    // Code  
}  
  
var i = 0;  
for ( ; i < 100; i++ ) {  
    // Code  
}  
  
var prop;  
for ( prop in object ) {  
    // Code  
}  
  
var object1 = {};  
var array1 = [];  
  
try {  
    // Code  
} catch ( e ) {  
    // Code  
} finally {  
    // Code  
}
```

# Javascript Style Guide

## # 함수 실행 ( How to execute Function )

- 익명함수의 실행 방법은 아래와 같다.
- **undefined** 를 parameter 로 받으면 **undefined** 이라는 변수로
- **undefined** 값을 적용할 수 있다.
- **Call** 과 **Apply** 차이는 아래와 같다.

```
(function(window, undefined) {  
  var a = null;  
  a = undefined;  
})(window);
```

```
var foo, boo,  
    context = this;  
  
foo = function( a, b ) {  
  ;  
  
  doo = function( a, b ) {  
    foo.call( context, a, b ); // 파라미터를 순서대로 넘김  
  
    foo.apply( context, [a, b] ); // 파라미터를 배열로 넘김  
  
    foo.apply( context, arguments ); // 해당 함수에 넘겨진 파라미터를 그대로 넘  
    김  
  };  
  
  doo( 1, "a" );
```

# Javascript Style Guide

## # 클로저 ( Closure )

아래와 같은 예로 함수 내 변수 선언은 중요하며 선언하지 않은 경우

- Global(window 개체) 변수로 선언되므로 주의해야 한다.
- 아래와 같은 상황을 방지하기 위해 “use strict”; 를 적절하게 사용한다.

```
var boo = 1;

(function(){
    var boo = 2;
    console.log( boo );// 2
})();

(function(){
    console.log( boo );// 1
})();

(function(){
    boo = 2;
    console.log( boo );// 2
})();

console.log( boo );// 2
```

```
(function(window, undefined) {

    "use strict";

    var a;

    // 이와 같은 방식은 undefined 값을 사용 가능
    if ( a == undefined ) {

    };

})(window);
```

# Javascript Style Guide

## # 클래스 객체 ( Class Object )

- 자바스크립트언어는 아직 java class 와 같은 개념이 없다.
- 이를 대체하기 위한 클래스 구현 방식은 **prototype** 방식과 **closure** 방식이 있다.
- Closure 를 이용한 클래스 구현 방식은 다음과 같다.
  - 외부에서 접근을 하지 못하는 private 변수 구현이 가능하다.

```
var ClassName = function() {  
    var _items = []; // private 변수  
  
    return {  
        _private: function() { // private 함수는 언더바(_)로 시작  
        },  
  
        add: function( item ) {  
            _items.push( item );  
        },  
  
        execute: function() {  
            console.log( 'items', 'count :', _items.length, 'contains :', _items );  
        }  
    };  
};  
  
var instance = new ClassName();  
instance.add('Andy');  
instance.add('Nina');  
instance.execute();
```

# Javascript Style Guide

## # 클래스 객체 ( Class Object )

- Prototype 방식을 이용한 클래스 구현 방식은 다음과 같다.
  - 변수의 접근을 public 하게 구현한다.
  - 생성시 마다 prototype 에 정의한 값은 모든 instance 에서 공유 되므로 주의해야 한다.

```
var ClassName = function() {  
    this.items = []; // public 변수  
};  
  
ClassName.prototype = {  
    _private: function() { // private 함수는 언더바(_)로 시작  
  
    },  
  
    add: function( item ) {  
        this.items.push( item );  
    },  
  
    execute: function() {  
        console.log( 'items', 'count :', this.items.length, 'contains :', this.items );  
    }  
};  
  
var instance = new ClassName();  
instance.add('Andy');  
Instance.a // 1  
instance.add('Nina');  
instance.execute();
```

# Javascript Style Guide

## # 함수로 파라미터 전달 방식

- 매개변수 방식
  - 순서가 정해져 있다.
  - 매개변수로 변수 할당시 함수를 변경해야 한다.
  - custom 한 확장시 arguments 값을 이용하는 경우도 있으나 어떤값이 될지 불명확하다.
- 혼합 리터럴 방식
  - 순서가 상관이 없다.
  - 매개변수 확장시 용이하다.

```
// 매개변수 방식
function foo( name, old, height ) {
  console.log( name + '은 ' + old + '살 이며 키는 ' + height + ' cm 입니다' );
}

// 혼합 리터럴 방식
function foo( option ) {
  var name = option.name,
      old = option.old,
      height = option.height;

  console.log( name + '은 ' + old + '살 이며 키는 ' + height + ' cm 입니다' );
}
```

## #실습

1. 서버정보
2. 전문명세
3. 퍼블리싱
4. 화면개발
  1. login
  2. 공지사항 목록
  3. 공지사항 상세
  4. intro



## #주의사항

### 1. 스택(Stack) Why? 메모리

```
/**
 * 사이드메뉴로 이동 시 중간 스택의 화면들을 삭제한다.
 */
function clearStack() {
    var isMovedFromSidemenu = dataManager.global('__MOVED_FROM_SIDEMENU__'),
        stackList;

    if ( isMovedFromSidemenu ) {
        stackList = M.info.stack();

        for ( var i = 1; i < stackList.length - 1; i ++ ) {
            M.page.remove(stackList[i].key);
        }

        dataManager.removeGlobal('__MOVED_FROM_SIDEMENU__');
    }
}
```



## #주의사항

### 2. Scroll

#### Native Scroll



#### iScroll



## #리소스 배포

리소스 관리 / 리소스 배포

### 리소스 업로드

App. 명 유라클 가이드 킷  파일 선택 선택된 파일 없음

등록 가능한 리소스 형식 png, html, js

#### 도움말

- ▶ 파일 선택 버튼을 클릭하여 리소스 파일을 선택합니다.
- ▶ 업로드 버튼을 클릭하면 팝업 화면으로 선택한 리소스 파일 정보
- ▶ 출력된 리소스를 확인 후 확인버튼을 클릭하면 리소스 파일에

### 리소스 관리

App. 명 유라클 가이드 킷

파일 상태 배포

기간 선택

기간 선택 2016.03.01  시  분 ~ 2016.03.04  시  분

<input type="checkbox"/>	파일명 △	파일경로	등록일	
<input type="checkbox"/>	bbd_dtl.js	res/www/js/ui/bbd_dtl.js	2016.03.02 10:49	<input type="button" value="상세"/>
<input type="checkbox"/>	envi_stu.js	res/www/js/ui/envi_stu.js	2016.03.02 09:28	<input type="button" value="상세"/>
<input type="checkbox"/>	login.js	res/www/js/ui/login.js	2016.03.02 09:28	<input type="button" value="상세"/>
<input type="checkbox"/>	menuManager.js	res/www/js/common/menuManager.js	2016.03.02 10:49	<input type="button" value="상세"/>
<input type="checkbox"/>	pushCommon.js	res/www/js/common/pushCommon.js	2016.03.02 10:49	<input type="button" value="상세"/>

## #App 배포관리

모바일 센터 > 앱 바이너리 버전 관리

### 앱 바이너리 버전 관리

App. 명  검색

모바일 서비스 목록

선택	App. 명	App ID	등록일
<input checked="" type="radio"/>	유라클 가이드 킷	kr.co.uracle.guidakit	2016.01.22

1

버전 목록 버전 등록

플랫폼유형	버전 ▾	등록일	
iOS	1.0.0	2016.01.22 14	<span>삭제</span>
안드로이드	1.0.0	2016.02.16 11	<span>삭제</span>

1

모바일 센터 > 앱 업데이트 관리

### 앱 업데이트 관리

App. 명  검색

모바일 서비스 목록

	App. 명	등록일	
<input checked="" type="radio"/>	유라클 가이드 킷	2016.01.22	<span>상세</span>

1

모바일 업데이트 목록 등록

필수업버전 ▾	디바이스유형	사용여부	
1.0.0	안드로이드	사용	<span>수정</span>

1

Q & A