**UserDefault**

实际上是一个xml文件

auto userDefault = UserDefault::getInstance();

userDefault->setBoolForKey("bool", true);

userDefault->setStringForKey("string", "test");

userDefault->setDoubleForKey("double", 12.0);

userDefault->setIntegerForKey("int", 100);

userDefault->setIntegerForKey("test", 200);

bool b1 = userDefault->getBoolForKey("bool");

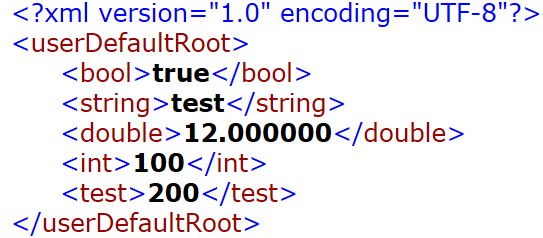
bool b2 = userDefault->getBoolForKey("bool1");

string str = userDefault->getStringForKey("string");

double d = userDefault->getDoubleForKey("double");

int i = userDefault->getIntegerForKey("int");

string path = userDefault->getXMLFilePath();



**FileUtils**

// 获取文本文件

virtual std::string getStringFromFile(const std::string& filename);

// 获取二进制文件

virtual Data getDataFromFile(const std::string& filename);

virtual unsigned char\* getFileDataFromZip(const std::string& zipFilePath, const std::string& filename, ssize\_t \*size);

virtual std::string fullPathForFilename(const std::string &filename) const;

属性：

std::string \_writablePath;

mutable std::unordered\_map<std::string, std::string> \_fullPathCache; // 全路径搜索缓存

std::string \_defaultResRootPath;

std::vector<std::string> \_searchPathArray; // 搜索路径

std::vector<std::string> \_searchResolutionsOrderArray; // 子搜索路径

ValueMap \_filenameLookupDict; // 名字字典，使用加密文件时适合用

**std::string FileUtils::fullPathForFilename(const std::string &filename) const**

{

if (filename.empty())

{

return "";

}

if (isAbsolutePath(filename))

{

return filename;

}

// Already Cached ?

auto cacheIter = \_fullPathCache.find(filename);

if(cacheIter != \_fullPathCache.end())

{

return cacheIter->second;

}

// Get the new file name.

const std::string newFilename( getNewFilename(filename) );

std::string fullpath;

for (const auto& searchIt : \_searchPathArray)

{

for (const auto& resolutionIt : \_searchResolutionsOrderArray)

{

fullpath = this->getPathForFilename(newFilename, resolutionIt, searchIt);

if (!fullpath.empty())

{

// Using the filename passed in as key.

\_fullPathCache.insert(std::make\_pair(filename, fullpath));

return fullpath;

}

}

}

if(isPopupNotify()){

CCLOG("cocos2d: fullPathForFilename: No file found at %s. Possible missing file.", filename.c\_str());

}

// The file wasn't found, return empty string.

return "";

}

fullPathForFilename（）搜索方式：

1、先从全路径缓存m\_fullPathCache中搜索

如果没有再按以下方式搜索：

1、 std::string newName = getNewFilename();

从m\_pFilenameLookupDict名字字典中查找替换名字。

2、组合路径：

路径1 + 子路径1 + 文件名

路径1 + 子路径2 + 文件名

路径1 + 子路径3 + 文件名

路径1 + 文件名

路径2 + 子路径1 + 文件名

路径2 + 子路径2 + 文件名

路径2 + 子路径3 + 文件名

路径2 + 文件名

。。。。

**SQLite**

实际上就是数据库操作，对应于LocalStorage