

# Pybind11

关于C++封装成.so 给Py调用的方式

本来一开始想考虑业务也许或者需要转换为Py调用的方式，方便后续的时候，现暂时搁置

现暂时写个文档用于方便后续使用

主要reference code为

[pybind11](#)

pybind11-cmake-example :在官方ReadMe.md 中有指向Rep里面有实例

环境依赖:

- python 使用环境
- python 开发环境 （包含头文件，动态库等） 头文件可用apt-get install python-dev ,使用安装
- pybind11 可直接git clone 官方代码
- cmake环境 cmake version>3.4

操作守则

- 需要有一定cmake基础，主要用于头文件和动态库或者静态库的链接

如下为带QT的动态库编译生成py.so方便调用的形式

**add\_subdirectory(pybind11)**该目录需要指向pybind11，在编译该流程的时候会自动寻找pybind11的文件，pybind11如果未编译，则会寻找pybind11相关依赖自己走编译流程。

```
cmake_minimum_required(VERSION 3.4...3.18)
project(example)
set(CMAKE_VERBOSE_MAKEFILE on)
set(CMAKE_CXX_COMPILER $ENV{CXX})
set(CMAKE_CXX_FLAGS "-O3 -std=c++11 -DNDEBUG")

set(Qt5_DIR "/usr/lib/x86_64-linux-gnu/cmake/Qt5" CACHE PATH "Directory that
contains Qt5.cmake")

find_package( OpenCV REQUIRED )
include_directories( ${OpenCV_INCLUDE_DIRS} )

add_subdirectory(pybind11)

find_package(Qt5 COMPONENTS Core Gui Widgets REQUIRED)

include_directories(./include ./)
link_directories(/usr/local/lib)
link_directories(${CMAKE_CURRENT_SOURCE_DIR}/lib)
#add_executable(pyexample
#    main.cpp
#    )
pybind11_add_module(example main.cpp)
set_target_properties(example PROPERTIES OUTPUT_NAME "example")
target_link_libraries(example PUBLIC Qt5::Widgets Qt5::Core Qt5::Gui cryptolib
${OpenCV_LIBS})
```

```
#install(TARGETS RUNTIME DESTINATION bin)
```

## main.cpp

```
#include <pybind11/pybind11.h>
namespace py = pybind11;

int add(int i, int j)
{
    return i + j;
}

PYBIND11_PLUGIN(example,m)
{
    py::module m("example", "pybind11 example plugin");
    m.def("add", &add, "A function which adds two numbers");
    return m.ptr();
}
```

编译流程，编译后会生成可调用python包

```
mkdir build
cd build
cmake ..
make
```

## 使用方式

```
> import example
> example.add(1,4)
> 5
```