// Copyright 2013 The Flutter Authors. All rights reserved.

// Use of this source code is governed by a BSD-style license that can be

// found in the LICENSE file.

#ifndef FLUTTER\_SHELL\_PLATFORM\_COMMON\_CLIENT\_WRAPPER\_INCLUDE\_FLUTTER\_METHOD\_RESULT\_FUNCTIONS\_H\_

#define FLUTTER\_SHELL\_PLATFORM\_COMMON\_CLIENT\_WRAPPER\_INCLUDE\_FLUTTER\_METHOD\_RESULT\_FUNCTIONS\_H\_

#include <functional>

#include <string>

#include <utility>

#include "method\_result.h"

namespace flutter {

class EncodableValue;

// Handler types for each of the MethodResult outcomes.

template <typename T>

using ResultHandlerSuccess = std::function<void(const T\* result)>;

template <typename T>

using ResultHandlerError = std::function<void(const std::string& error\_code,

const std::string& error\_message,

const T\* error\_details)>;

template <typename T>

using ResultHandlerNotImplemented = std::function<void()>;

// An implementation of MethodResult that pass calls through to provided

// function objects, for ease of constructing one-off result handlers.

template <typename T = EncodableValue>

class MethodResultFunctions : public MethodResult<T> {

public:

// Creates a result object that calls the provided functions for the

// corresponding MethodResult outcomes.

MethodResultFunctions(ResultHandlerSuccess<T> on\_success,

ResultHandlerError<T> on\_error,

ResultHandlerNotImplemented<T> on\_not\_implemented)

: on\_success\_(on\_success),

on\_error\_(on\_error),

on\_not\_implemented\_(std::move(on\_not\_implemented)) {}

virtual ~MethodResultFunctions() = default;

// Prevent copying.

MethodResultFunctions(MethodResultFunctions const&) = delete;

MethodResultFunctions& operator=(MethodResultFunctions const&) = delete;

protected:

// |flutter::MethodResult|

void SuccessInternal(const T\* result) override {

if (on\_success\_) {

on\_success\_(result);

}

}

// |flutter::MethodResult|

void ErrorInternal(const std::string& error\_code,

const std::string& error\_message,

const T\* error\_details) override {

if (on\_error\_) {

on\_error\_(error\_code, error\_message, error\_details);

}

}

// |flutter::MethodResult|

void NotImplementedInternal() override {

if (on\_not\_implemented\_) {

on\_not\_implemented\_();

}

}

private:

ResultHandlerSuccess<T> on\_success\_;

ResultHandlerError<T> on\_error\_;

ResultHandlerNotImplemented<T> on\_not\_implemented\_;

};

} // namespace flutter

#endif // FLUTTER\_SHELL\_PLATFORM\_COMMON\_CLIENT\_WRAPPER\_INCLUDE\_FLUTTER\_METHOD\_RESULT\_FUNCTIONS\_H\_