#pragma once

#include <twml/defines.h>

#ifdef \_\_cplusplus

#include <stddef.h>

#include <stdexcept>

#include <stdint.h>

#include <string>

namespace twml {

class Error : public std::runtime\_error {

private:

twml\_err m\_err;

public:

Error(twml\_err err, const std::string &msg) :

std::runtime\_error(msg), m\_err(err)

{

}

twml\_err err() const

{

return m\_err;

}

};

class ThriftInvalidField: public twml::Error {

public:

ThriftInvalidField(int16\_t field\_id, const std::string& func) :

Error(TWML\_ERR\_THRIFT,

"Found invalid field (" + std::to\_string(field\_id)

+ ") while reading thrift [" + func + "]")

{

}

};

class ThriftInvalidType: public twml::Error {

public:

ThriftInvalidType(uint8\_t type\_id, const std::string& func, const std::string type) :

Error(TWML\_ERR\_THRIFT,

"Found invalid type (" + std::to\_string(type\_id) +

") while reading thrift [" + func + "::" + type + "]")

{

}

};

}

#endif