#pragma once

#include <twml/Error.h>

#include <iostream>

#define HANDLE\_EXCEPTIONS(fn) do { \

try { \

fn \

} catch(const twml::Error &e) { \

std::cerr << e.what() << std::endl; \

return e.err(); \

} catch(...) { \

std::cerr << "Unknown error\n"; \

return TWML\_ERR\_UNKNOWN; \

} \

} while(0)

#define TWML\_CHECK(fn, msg) do { \

twml\_err err = fn; \

if (err == TWML\_ERR\_NONE) break; \

throw twml::Error(err, msg); \

} while(0)

#define CHECK\_THRIFT\_TYPE(real\_type, expected\_type, type) do { \

int real\_type\_val = real\_type; \

if (real\_type\_val != expected\_type) { \

throw twml::ThriftInvalidType(real\_type\_val, \_\_func\_\_, type); \

} \

} while(0)