// Licensed to the Apache Software Foundation (ASF) under one

// or more contributor license agreements. See the NOTICE file

// distributed with this work for additional information

// regarding copyright ownership. The ASF licenses this file

// to you under the Apache License, Version 2.0 (the

// "License"); you may not use this file except in compliance

// with the License. You may obtain a copy of the License at

//

// http://www.apache.org/licenses/LICENSE-2.0

//

// Unless required by applicable law or agreed to in writing,

// software distributed under the License is distributed on an

// "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY

// KIND, either express or implied. See the License for the

// specific language governing permissions and limitations

// under the License.

#include "csv.h"

#include <memory>

#include "arrow/python/common.h"

namespace arrow {

using csv::InvalidRow;

using csv::InvalidRowHandler;

using csv::InvalidRowResult;

namespace py {

namespace csv {

InvalidRowHandler MakeInvalidRowHandler(PyInvalidRowCallback cb, PyObject\* py\_handler) {

if (cb == nullptr) {

return InvalidRowHandler{};

}

struct Handler {

PyInvalidRowCallback cb;

std::shared\_ptr<OwnedRefNoGIL> handler\_ref;

InvalidRowResult operator()(const InvalidRow& invalid\_row) {

InvalidRowResult result;

auto st = SafeCallIntoPython([&]() -> Status {

result = cb(handler\_ref->obj(), invalid\_row);

if (PyErr\_Occurred()) {

PyErr\_WriteUnraisable(handler\_ref->obj());

}

return Status::OK();

});

ARROW\_UNUSED(st);

return result;

}

};

Py\_INCREF(py\_handler);

return Handler{cb, std::make\_shared<OwnedRefNoGIL>(py\_handler)};

}

} // namespace csv

} // namespace py

} // namespace arrow