

TotalViewITCH.jl

Data Storage

- Store processed data to CSV

```
struct FileSystem <: Backend
    url
    headers::Dict{Symbol,Vector{String}}
end
```

- or MongoDB

```
struct MongoDB <: Backend
    url
    db_name
    nlevels
end
```

TotalViewITCH.jl on master [!\$] via 1.10.4 via base
→ tree data/test

```
data/test
├── messages
│   ├── ticker=A
│   │   └── date=2013-03-14
│   │       └── partition.csv
│   └── ticker=B
│       └── date=2013-03-14
│           └── partition.csv
├── noii
│   └── date=2013-03-14
│       └── partition.csv
├── orderbooks
│   ├── ticker=A
│   │   └── date=2013-03-14
│   │       └── partition.csv
│   └── ticker=B
│       └── date=2013-03-14
│           └── partition.csv
└── trades
    ├── ticker=A
    │   └── date=2013-03-14
    │       └── partition.csv
    └── ticker=B
        └── date=2013-03-14
            └── partition.csv
```

17 directories, 7 files

TotalViewITCH.jl

Data Storage

- Need a different data format?
- Julia's generics and abstract types to the rescue!

```
const Writable = Union{Book,OrderMessage,TradeMessage,NOIIMessage}

abstract type Backend end

function ping(b::Backend) end
function check_exists(date::Date, ticker::String, b::Backend)::Bool end
function build(b::Backend)::Bool end
function insert(b::Backend, items, collection, ticker, date)::Int end
function find(b::Backend, collection, ticker, date) end
function clean(date::Date, ticker::String, b::Backend)::Bool end
function clean(date::Date, b::Backend)::Bool end
function clean(ticker::String, b::Backend)::Bool end
function teardown(b::Backend)::Bool end
```