# StackTraces

`Base.StackTraces.StackFrame` — Type

```
StackFrame
```

Stack information representing execution context, with the following fields:

- `func::Symbol`

  The name of the function containing the execution context.
- `linfo::Union{Core.MethodInstance, CodeInfo, Nothing}`

  The MethodInstance containing the execution context (if it could be found).
- `file::Symbol`

  The path to the file containing the execution context.
- `line::Int`

  The line number in the file containing the execution context.
- `from_c::Bool`

  True if the code is from C.
- `inlined::Bool`

  True if the code is from an inlined frame.
- `pointer::UInt64`

  Representation of the pointer to the execution context as returned by `backtrace`.

`Base.StackTraces.StackTrace` — Type

```
StackTrace
```

An alias for `Vector{StackFrame}` provided for convenience; returned by calls to `stacktrace`.

### `Base.StackTraces.stacktrace` — Function

```
stacktrace([trace::Vector{Ptr{Cvoid}},] [c_funcs::Bool=false]) -> StackTrace
```

Returns a stack trace in the form of a vector of `StackFrame`s. (By default stacktrace doesn't return C functions, but this can be enabled.) When called without specifying a trace, `stacktrace` first calls `backtrace`.

The following methods and types in `Base.StackTraces` are not exported and need to be called e.g. as `StackTraces.lookup(ptr)`.

### `Base.StackTraces.lookup` — Function

```
lookup(pointer::Ptr{Cvoid}) -> Vector{StackFrame}
```

Given a pointer to an execution context (usually generated by a call to `backtrace`), looks up stack frame context information. Returns an array of frame information for all functions inlined at that point, innermost function first.

### `Base.StackTraces.remove_frames!` — Function

```
remove_frames!(stack::StackTrace, name::Symbol)
```

Takes a `StackTrace` (a vector of `StackFrames`) and a function name (a `Symbol`) and removes the `StackFrame` specified by the function name from the `StackTrace` (also removing all frames above the specified function). Primarily used to remove `StackTraces` functions from the `StackTrace` prior to returning it.

```
remove_frames!(stack::StackTrace, m::Module)
```

Returns the `StackTrace` with all `StackFrame`s from the provided `Module` removed.

« C Standard Library SIMD Support »

Powered by Documenter.jl and the Julia Programming Language.