Crate proc_macro

Since 1.15.0 ·

Settings Help Summary

A support library for macro authors when defining new macros.

This library, provided by the standard distribution, provides the types consumed in the interfaces of procedurally defined macro definitions such as function-like macros `#[proc_macro]`, macro attributes `#[proc_macro_attribute]` and custom derive attributes `#[proc_macro_derive]`.

See the book for more.

Modules

token_stream	Public implementation details for the `TokenStream` type, such as iterators.
tracked_env Experimental	Tracked access to environment variables.
tracked_path Experimental	Tracked access to additional files.

Macros

quote Experimental	`quote!(..)` accepts arbitrary tokens and expands into a `TokenStream` describing the input. For example, `quote!(a + b)` will produce an expression, that, when evaluated, constructs the `TokenStream [Ident("a"), Punct('+', Alone), Ident("b")]`.

Structs

Group	A delimited token stream.
Ident	An identifier (`ident`).
LexError	Error returned from `TokenStream::from_str`.
Literal	A literal string (`"hello"`), byte string (`b"hello"`), C string (`c"hello"`), character (`'a'`), byte character (`b'a'`), an integer or floating point number with or without a suffix (`1`, `1u8`, `2.3`, `2.3f32`). Boolean literals like `true` and `false` do not belong here, they are `Idents`.
Punct	A `Punct` is a single punctuation character such as `+`, `-` or `#`.
Span	A region of source code, along with macro expansion information.

TokenStream	The main type provided by this crate, representing an abstract stream of tokens, or, more specifically, a sequence of token trees. The type provides interfaces for iterating over those token trees and, conversely, collecting a number of token trees into one stream.
Diagnostic Experimental	A structure representing a diagnostic message and associated children messages.
ExpandError Experimental	Error returned from `TokenStream::expand_expr`.
SourceFile Experimental	The source file of a given `Span`.

Enums

Delimiter	Describes how a sequence of token trees is delimited.
Spacing	Indicates whether a `Punct` token can join with the following token to form a multi-character operator.
TokenTree	A single token or a delimited sequence of token trees (e.g., `[1, (), ..]`).
Level Experimental	An enum representing a diagnostic level.

Traits

MultiSpan Experimental	Trait implemented by types that can be converted into a set of `Span`s.
ToTokens Experimental	Types that can be interpolated inside a `quote!` invocation.

Functions

is_available	Determines whether proc_macro has been made accessible to the currently running program.
quote Experimental	Quote a `TokenStream` into a `TokenStream`. This is the actual implementation of the `quote!()` proc macro.
quote_span Experimental	Quote a `Span` into a `TokenStream`. This is needed to implement a custom quoter.