import {SourceSpan} from './source\_span';

export {SourceLocation} from './source\_location';

export {SourceSpan} from './source\_span';

/\*\*

\* An object that can be passed to [[LegacySharedOptions.logger]] to control how

\* Sass emits warnings and debug messages.

\*

\* @example

\*

\* ```js

\* const fs = require('fs');

\* const sass = require('sass');

\*

\* let log = "";

\* sass.renderSync({

\* file: 'input.scss',

\* logger: {

\* warn(message, options) {

\* if (options.span) {

\* log += `${span.url}:${span.start.line}:${span.start.column}: ` +

\* `${message}\n`;

\* } else {

\* log += `::: ${message}\n`;

\* }

\* }

\* }

\* });

\*

\* fs.writeFileSync('log.txt', log);

\* ```

\*

\* @category Logger

\*/

export interface Logger {

/\*\*

\* This method is called when Sass emits a warning, whether due to a [`@warn`

\* rule](https://sass-lang.com/documentation/at-rules/warn) or a warning

\* generated by the Sass compiler.

\*

\* If this is `undefined`, Sass will print warnings to standard error.

\*

\* @param message - The warning message.

\* @param options.deprecation - Whether this is a deprecation warning.

\* @param options.span - The location in the Sass source code that generated this

\* warning.

\* @param options.stack - The Sass stack trace at the point the warning was issued.

\*/

warn?(

message: string,

options: {

deprecation: boolean;

span?: SourceSpan;

stack?: string;

}

): void;

/\*\*

\* This method is called when Sass emits a debug message due to a [`@debug`

\* rule](https://sass-lang.com/documentation/at-rules/debug).

\*

\* If this is `undefined`, Sass will print debug messages to standard error.

\*

\* @param message - The debug message.

\* @param options.span - The location in the Sass source code that generated this

\* debug message.

\*/

debug?(message: string, options: {span: SourceSpan}): void;

}

/\*\*

\* A namespace for built-in [[Logger]]s.

\*

\* @category Logger

\* @compatibility dart: "1.43.0", node: false

\*/

export namespace Logger {

/\*\*

\* A [[Logger]] that silently ignores all warnings and debug messages.

\*

\* @example

\*

\* ```js

\* const sass = require('sass');

\*

\* const result = sass.renderSync({

\* file: 'input.scss',

\* logger: sass.Logger.silent,

\* });

\* ```

\*/

export const silent: Logger;

}