

dotnamecpp::utils:: IDirectoryManager
<ul style="list-style-type: none"> <li>+ virtual ~IDirectoryManager() =default</li> <li>+ virtual Result&lt; void, FileError &gt; createDirectory(const std::filesystem::path &amp;dirPath) const =0</li> <li>+ virtual Result&lt; void, FileError &gt; removeDirectory(const std::filesystem::path &amp;dirPath) const =0</li> <li>+ virtual Result&lt; std::uintmax_t, FileError &gt; removeDirectoryRecursive(const std::filesystem::path &amp;dirPath) const =0</li> <li>+ virtual bool exists(const std::filesystem::path &amp;dirPath) const =0</li> <li>+ virtual Result&lt; bool, FileError &gt; isEmpty(const std::filesystem::path &amp;dirPath) const =0</li> <li>+ virtual Result&lt; std::vector&lt; std::filesystem::path &gt;, FileError &gt; listEntries(const std::filesystem::path &amp;dirPath) const =0</li> <li>+ virtual Result&lt; std::vector&lt; std::filesystem::path &gt;, FileError &gt; listEntriesRecursive(const std::filesystem::path &amp;dirPath) const =0</li> <li>+ virtual Result&lt; std::filesystem::path, FileError &gt; getCurrentDirectory() const =0</li> <li>+ virtual Result&lt; std::filesystem::path, FileError &gt; getTempDirectory() const =0</li> </ul>



dotnamecpp::utils:: DirectoryManager
<ul style="list-style-type: none"> <li>+ DirectoryManager() =default</li> <li>+ DirectoryManager(const DirectoryManager &amp;)=delete</li> <li>+ DirectoryManager &amp; operator=(const DirectoryManager &amp;)=delete</li> <li>+ DirectoryManager(DirectoryManager &amp;&amp;)=delete</li> <li>+ DirectoryManager &amp; operator=(DirectoryManager &amp;&amp;)=delete</li> <li>+ ~DirectoryManager() override=default</li> <li>+ Result&lt; void, FileError &gt; createDirectory(const std::filesystem::path &amp;dirPath) const override</li> <li>+ Result&lt; void, FileError &gt; removeDirectory(const std::filesystem::path &amp;dirPath) const override</li> <li>+ Result&lt; std::uintmax_t, FileError &gt; removeDirectoryRecursive(const std::filesystem::path &amp;dirPath) const override</li> <li>+ bool exists(const std::filesystem::path &amp;dirPath) const override</li> <li>+ Result&lt; bool, FileError &gt; isEmpty(const std::filesystem::path &amp;dirPath) const override</li> <li>+ Result&lt; std::vector&lt; std::filesystem::path &gt;, FileError &gt; listEntries(const std::filesystem::path &amp;dirPath) const override</li> <li>+ Result&lt; std::vector&lt; std::filesystem::path &gt;, FileError &gt; listEntriesRecursive(const std::filesystem::path &amp;dirPath) const override</li> <li>+ Result&lt; std::filesystem::path, FileError &gt; getCurrentDirectory() const override</li> <li>+ Result&lt; std::filesystem::path, FileError &gt; getTempDirectory() const override</li> </ul>