package com.twitter.product\_mixer.core.functional\_component.side\_effect

import com.twitter.product\_mixer.core.model.common.Component

import com.twitter.product\_mixer.core.model.common.identifier.SideEffectIdentifier

import com.twitter.stitch.Stitch

/\*\*

\* A side-effect is a ancillary action that doesn't affect the result of execution directly.

\*

\* For example: Logging, history stores

\*

\* Implementing components can express failures by throwing an exception. These exceptions

\* will be caught and not affect the request processing.

\*

\* @note Side effects execute asynchronously in a fire-and-forget way, it's important to add alerts

\* to the [[SideEffect]] component itself since a failures wont show up in metrics

\* that just monitor your pipeline as a whole.

\*

\* @see [[ExecuteSynchronously]] for modifying a [[SideEffect]] to execute with synchronously with

\* the request waiting on the side effect to complete, this will impact the overall request's latency

\*\*/

trait SideEffect[-Inputs] extends Component {

/\*\* @see [[SideEffectIdentifier]] \*/

override val identifier: SideEffectIdentifier

def apply(inputs: Inputs): Stitch[Unit]

}