



Bundesverwaltungsamt



IsyFact-Standard

Nutzungskonzept isy-util

Version 1.5
27.03.2015



„ des Bundesverwaltungsamts ist lizenziert unter einer Creative Commons Namensnennung 4.0 International Lizenz.



„Nutzungskonzept plis-util“
des Bundesverwaltungsamts ist lizenziert unter einer
Creative Commons Namensnennung 4.0 International Lizenz.

Die Lizenzbestimmungen können unter folgender URL heruntergeladen
werden: <http://creativecommons.org/licenses/by/4.0>

Ansprechpartner:

Referat Z II 2
Bundesverwaltungsamt
E-Mail: isyfact@bva.bund.de
Internet: www.isyfact.de

Dokumentinformationen

Dokumenten-ID:	Nutzungskonzept isy-util.docx
----------------	-------------------------------

Inhaltsverzeichnis

1.	Einleitung.....	5
2.	Aufbau von isy-util.....	6
2.1.	Annotationen.....	6
2.2.	Interzeptoren.....	6
2.3.	Common.....	6
2.4.	Package exception.....	6
2.5.	Package monitoring	7
2.6.	Package spring.....	7

1. Einleitung

Die Bibliothek isy-util bietet nützliche Hilfsmittel, die von den Anwendungen der IsyFact genutzt werden können. Es handelt sich dabei um kleinere Utility-Klassen, welche die Implementierung vereinfachen. Diese werden im Folgenden überblicksartig beschrieben. Details sind den JavaDoc der einzelnen Klassen zu entnehmen.

2. Aufbau von isy-util

2.1. Annotationen

Im Package `de.bund.bva.pliscommon.util.annotations` der Bibliothek isy-util befinden sich Annotationen, die von Anwendungen der IsyFact genutzt werden können, um Klassen mit bestimmten Eigenschaften zu kennzeichnen.

Aktuell sind die folgenden Annotationen in dem Package enthalten:

- `@DataAccessObject`: Diese Annotation kennzeichnet eine Klasse als DAO-Implementierungsklasse.
- `@KomponentenFassade`: Die Annotation kennzeichnet eine Komponenten-Fassadenklasse, die alle Interfaces einer Komponente implementiert und an die tatsächlichen Implementierungsklassen delegiert.
- `@LaufzeitLogging`: Mit dieser Annotation werden Methoden gekennzeichnet, deren Laufzeit gemessen und geloggt werden soll.

2.2. Interzeptoren

Im Package `de.bund.bva.pliscommon.util.aop` bietet die Bibliothek hilfreiche Interzeptoren an:

- `LoggingTraceInterceptor`: Dieser Trace-Interceptor erlaubt es, aufgerufene Methoden mit ihren Parametern in das Trace-Log zu schreiben.
- `LogTimingInterceptor`: Mit diesem Interceptor wird ein Methodenaufruf inklusive Laufzeit geloggt.
- `PlisTraceInterceptor`: Dieser Trace-Interceptor gibt den Beginn, das Ende und die Dauer eines Methodenaufrufs aus.

2.3. Common

Das Package `de.bund.bva.pliscommon.util.common` enthält allgemeine Klassen, die keiner speziellen Funktionalität zugeordnet sind. Zurzeit besteht sie aus folgender Klasse:

- `RecursiveToStringBuilder`: Diese Klasse erzeugt eine Textausgabe für Objekte, die keine geeignete `toString`-Methode implementieren.

2.4. Package exception

In `de.bund.bva.pliscommon.util.exception` sind Klassen enthalten, die für die Fehlerbehandlung einzusetzen sind:

- `MessageSourceFehlertextProvider`: Durch diese Klasse wird der `FehlertextProvider` implementiert, der sich im Package `de.bund.bva.pliscommon.exception` der Bibliothek `isy-exception-core` befindet. Der `FehlertextProvider` beschreibt Methoden zum Auslesen von Fehlertexten auf Basis von IDs der Ausnahmen und Parameter, sofern welche vorhanden sind.

2.5. Package monitoring

Hilfswerkzeuge für Performance-Messungen liegen in dem Package `de.bund.bva.pliscommon.util.monitoring`. Folgendes Werkzeug ist in dem Package enthalten:

- `PerformanceMonitor`: Der `PerformanceMonitor` erlaubt es, Laufzeiten in Anwendungen zu messen. Dabei wird die Performance pro Request gemessen. Es kann eingestellt werden, wann der Beginn und das Ende eines Requests liegen sollen. Weiterhin können Teilen eines Requests Tags zugeordnet werden, um sie gesondert zu messen. Beispielsweise sind Tags für eine gesonderte Messung von Datenbankzugriffen sinnvoll.

2.6. Package spring

Das Package `de.bund.bva.pliscommon.util.spring` enthält Werkzeuge für den Umgang mit Spring:

- `MessageSourceHolder`: Der `MessageSourceHolder` ist eine Referenz auf die aktuelle `MessageSource`-Bean. Zusätzlich bietet er Convenience-Funktionen zum Auslesen von `ResourceBundle`-Einträgen aus der `MessageSource`.