Adaptive Monitoring

Björn Weißenfels

11. Oktober 2012

Gliederung

1. Einleitung
   - Motivation
   - Ansatz
   - Syntax
2. ProbeController
   - API
   - ProbeController
3. Konfiguration über Datei
   - Kieker Properties
   - Konfigurationsdatei
4. Live Demo

Gliederung

1. Einleitung
   - Motivation
   - Ansatz
   - Syntax
2. ProbeController
   - API
   - ProbeController
3. Konfiguration über Datei
   - Kieker Properties
   - Konfigurationsdatei
4. Live Demo

Motivation

- (De)Aktivierung des Aufzeichnens von Methoden zur Laufzeit.
- Über Methoden in ProbeController.
- Über externe Konfigurationsdatei (ähnlich aop.xml).

Motivation

- (De)Aktivierung des Aufzeichnens von Methoden zur Laufzeit.
- Über Methoden in ProbeController.
- Über externe Konfigurationsdatei (ähnlich aop.xml).

Motivation

- (De)Aktivierung des Aufzeichnens von Methoden zur Laufzeit.
- Über Methoden in ProbeController.
- Über externe Konfigurationsdatei (ähnlich aop.xml).

Ansatz

- Liste von Pattern mit Signaturen vergleichen
- Idee: Code von aspectj wiederverwenden
- leider zu komplex
- Daher: Eigene Syntax und Implementierung

Ansatz

- Liste von Pattern mit Signaturen vergleichen
- Idee: Code von aspectj wiederverwenden
- leider zu komplex
- Daher: Eigene Syntax und Implementierung

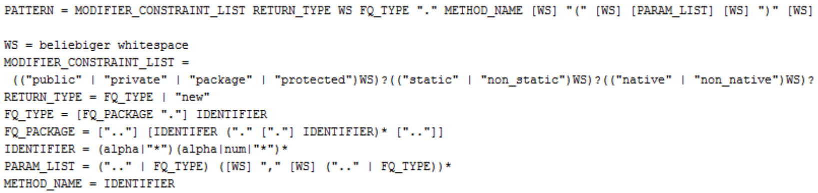
Ansatz

- Liste von Pattern mit Signaturen vergleichen
- Idee: Code von aspectj wiederverwenden
- leider zu komplex
- Daher: Eigene Syntax und Implementierung

Ansatz

- Liste von Pattern mit Signaturen vergleichen
- Idee: Code von aspectj wiederverwenden
- leider zu komplex
- Daher: Eigene Syntax und Implementierung

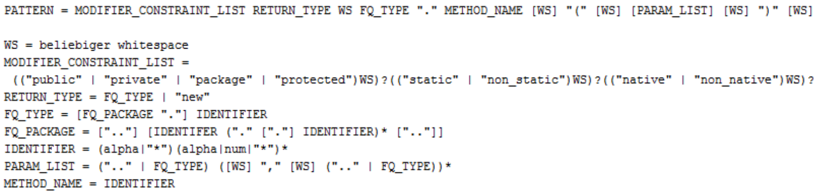
Syntax



```
PATTERN = MODIFIER_CONSTRAINT_LIST RETURN_TYPE WS FQ_TYPE "." METHOD_NAME [WS] "(" [WS] [PARAM_LIST] [WS] ")" [WS]

WS = beliebiger whitespace
MODIFIER_CONSTRAINT_LIST =
 (("public" | "private" | "package" | "protected")WS)?(("static" | "non_static")WS)?(("native" | "non_native")WS)?
RETURN_TYPE = FQ_TYPE | "new"
FQ_TYPE = [FQ_PACKAGE "."] IDENTIFIER
FQ_PACKAGE = [".."] [IDENTIFER ("." ["."] IDENTIFIER)* [".."]]
IDENTIFIER = (alpha|"*")(alpha|num|"*")*
PARAM_LIST = (".." | FQ_TYPE) ([WS] "," [WS] (".." | FQ_TYPE))*
METHOD_NAME = IDENTIFIER
```

- *
- ..* ..*.*(..)
- public non_static ..* package..Class.get*()
- int ..*.*(..)

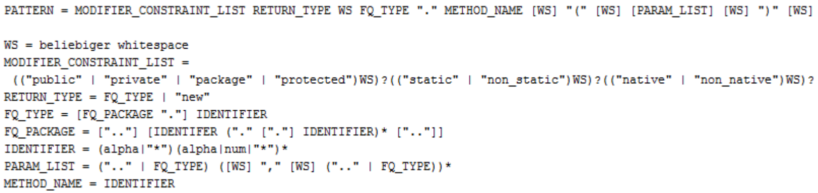
Syntax



```
PATTERN = MODIFIER_CONSTRAINT_LIST RETURN_TYPE WS FQ_TYPE "." METHOD_NAME [WS] "(" [WS] [PARAM_LIST] [WS] ")" [WS]

WS = beliebiger whitespace
MODIFIER_CONSTRAINT_LIST =
 (("public" | "private" | "package" | "protected")WS)?(("static" | "non_static")WS)?(("native" | "non_native")WS)?
RETURN_TYPE = FQ_TYPE | "new"
FQ_TYPE = [FQ_PACKAGE "."] IDENTIFIER
FQ_PACKAGE = [".."] [IDENTIFER ("." ["."] IDENTIFIER)* [".."]]
IDENTIFIER = (alpha|"*")(alpha|num|"*")*
PARAM_LIST = (".." | FQ_TYPE) ([WS] "," [WS] (".." | FQ_TYPE))*
METHOD_NAME = IDENTIFIER
```

- *
- ..* ..*.*(..)
- public non_static ..* package..Class.get*()
- int ..*.*(..)

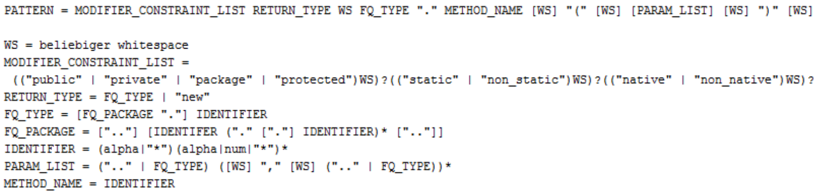
Syntax



```
PATTERN = MODIFIER_CONSTRAINT_LIST RETURN_TYPE WS FQ_TYPE "." METHOD_NAME [WS] "(" [WS] [PARAM_LIST] [WS] ")" [WS]

WS = beliebiger whitespace
MODIFIER_CONSTRAINT_LIST =
 (("public" | "private" | "package" | "protected")WS)?(("static" | "non_static")WS)?(("native" | "non_native")WS)?
RETURN_TYPE = FQ_TYPE | "new"
FQ_TYPE = [FQ_PACKAGE "."] IDENTIFIER
FQ_PACKAGE = [".."] [IDENTIFER ("." ["."] IDENTIFIER)* [".."]]
IDENTIFIER = (alpha|"*")(alpha|num|"*")*
PARAM_LIST = (".." | FQ_TYPE) ([WS] "," [WS] (".." | FQ_TYPE))*
METHOD_NAME = IDENTIFIER
```

- *
- ..* ..*.*(..)
- public non_static ..* package..Class.get*()
- int ..*.*(..)

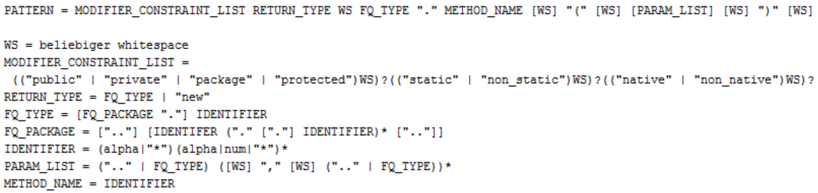
Syntax



```
PATTERN = MODIFIER_CONSTRAINT_LIST RETURN_TYPE WS FQ_TYPE "." METHOD_NAME [WS] "(" [WS] [PARAM_LIST] [WS] ")" [WS]

WS = beliebiger whitespace
MODIFIER_CONSTRAINT_LIST =
 (("public" | "private" | "package" | "protected")WS)?(("static" | "non_static")WS)?(("native" | "non_native")WS)?
RETURN_TYPE = FQ_TYPE | "new"
FQ_TYPE = [FQ_PACKAGE "."] IDENTIFIER
FQ_PACKAGE = [".."] [IDENTIFER ("." ["."] IDENTIFIER)* [".."]]
IDENTIFIER = (alpha|"*")(alpha|num|"*")*
PARAM_LIST = (".." | FQ_TYPE) ([WS] "," [WS] (".." | FQ_TYPE))*
METHOD_NAME = IDENTIFIER
```

- *
- ..* ..*.*(..)
- public non_static ..* package..Class.get*()
- int ..*.*(..)

Syntax



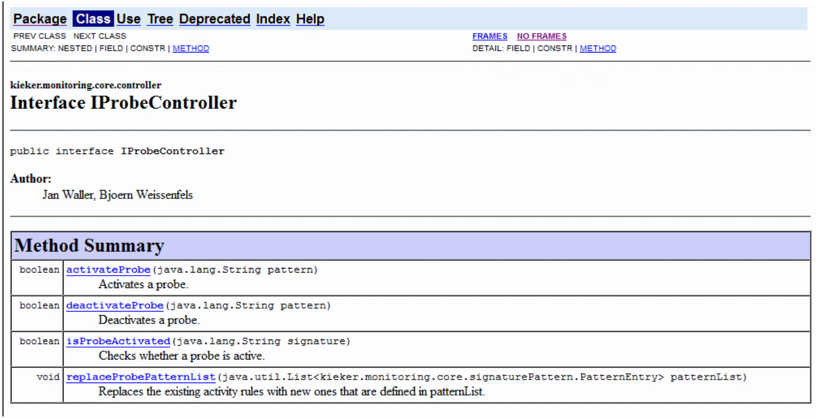
```
PATTERN = MODIFIER_CONSTRAINT_LIST RETURN_TYPE WS FQ_TYPE "." METHOD_NAME [WS] "(" [WS] [PARAM_LIST] [WS] ")" [WS]

WS = beliebiger whitespace
MODIFIER_CONSTRAINT_LIST =
 (("public" | "private" | "package" | "protected")WS)?(("static" | "non_static")WS)?(("native" | "non_native")WS)?
RETURN_TYPE = FQ_TYPE | "new"
FQ_TYPE = [FQ_PACKAGE "."] IDENTIFIER
FQ_PACKAGE = [".."] [IDENTIFER ("." ["."] IDENTIFIER)* [".."]]
IDENTIFIER = (alpha|"*")(alpha|num|"*")*
PARAM_LIST = (".." | FQ_TYPE) ([WS] "," [WS] (".." | FQ_TYPE))*
METHOD_NAME = IDENTIFIER
```

- *
- ..* ..*.*(..)
- public non_static ..* package..Class.get*()
- int ..*.*(..)

Gliederung

1. Einleitung
   - Motivation
   - Ansatz
   - Syntax
2. ProbeController
   - API
   - ProbeController
3. Konfiguration über Datei
   - Kieker Properties
   - Konfigurationsdatei
4. Live Demo

IProbeController



Package Class Use Tree Deprecated Index Help
PREV CLASS NEXT CLASS
FRAMES NO FRAMES
SUMMARY: NESTED | FIELD | CONSTR | METHOD
DETAIL: FIELD | CONSTR | METHOD

kieker.monitoring.core.controller

Interface IProbeController

```
public interface IProbeController
```

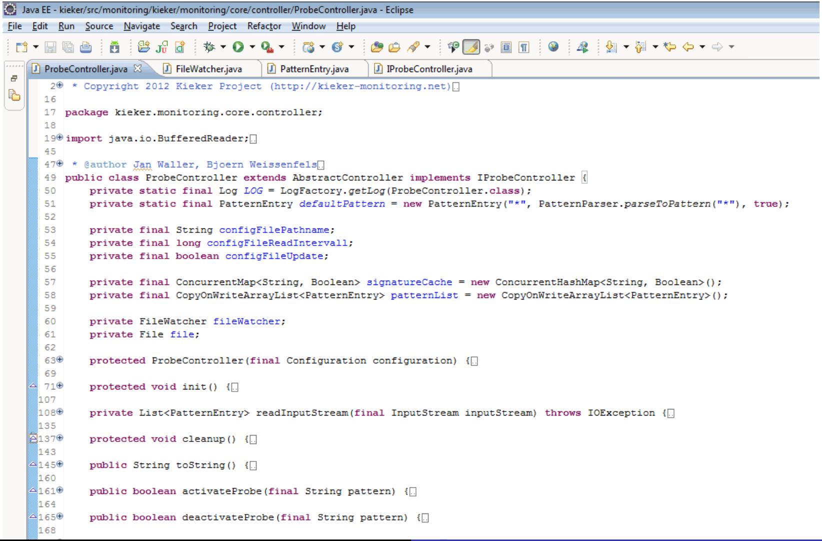
Author:

Jan Waller, Bjoern Weissenfels

Method Summary

boolean	`activateProbe(java.lang.String pattern)` Activates a probe.
boolean	`deactivateProbe(java.lang.String pattern)` Deactivates a probe.
boolean	`isProbeActivated(java.lang.String signature)` Checks whether a probe is active.
void	`replaceProbePatternList(java.util.List<kieker.monitoring.core.signaturePattern.PatternEntry> patternList)` Replaces the existing activity rules with new ones that are defined in patternList.

ProbeController

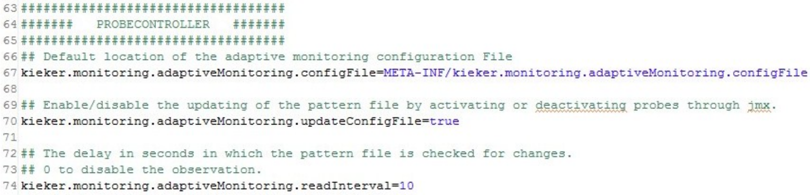

Java EE - kieker/src/monitoring/kieker/monitoring/core/controller/ProbeController.java - Eclipse
File Edit Run Source Navigate Search Project Refactor Window Help
ProbeController.java
FileWatcher.java
PatternEntry.java
IProbeController.java

```
 * Copyright 2012 Kieker Project (http://kieker-monitoring.net)□

package kieker.monitoring.core.controller;

import java.io.BufferedReader;□

 * @author Jan Waller, Bjoern Weissenfels□
public class ProbeController extends AbstractController implements IProbeController {
    private static final Log LOG = LogFactory.getLog(ProbeController.class);
    private static final PatternEntry defaultPattern = new PatternEntry("*", PatternParser.parseToPattern("*"), true);

    private final String configFilePathname;
    private final long configFileReadIntervall;
    private final boolean configFileUpdate;

    private final ConcurrentMap<String, Boolean> signatureCache = new ConcurrentHashMap<String, Boolean>();
    private final CopyOnWriteArrayList<PatternEntry> patternList = new CopyOnWriteArrayList<PatternEntry>();

    private FileWatcher fileWatcher;
    private File file;

    protected ProbeController(final Configuration configuration) {□

    protected void init() {□

    private List<PatternEntry> readInputStream(final InputStream inputStream) throws IOException {□

    protected void cleanup() {□

    public String toString() {□

    public boolean activateProbe(final String pattern) {□

    public boolean deactivateProbe(final String pattern) {□
```

Gliederung

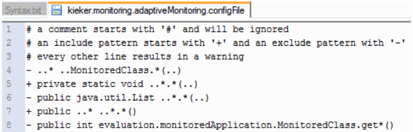
1. Einleitung
   - Motivation
   - Ansatz
   - Syntax
2. ProbeController
   - API
   - ProbeController
3. Konfiguration über Datei
   - Kieker Properties
   - Konfigurationsdatei
4. Live Demo

Kieker Properties



```
###################################
#######   PROBECONTROLLER   #######
###################################
## Default location of the adaptive monitoring configuration File
kieker.monitoring.adaptiveMonitoring.configFile=META-INF/kieker.monitoring.adaptiveMonitoring.configFile

## Enable/disable the updating of the pattern file by activating or deactivating probes through jmx.
kieker.monitoring.adaptiveMonitoring.updateConfigFile=true

## The delay in seconds in which the pattern file is checked for changes.
## 0 to disable the observation.
kieker.monitoring.adaptiveMonitoring.readInterval=10
```

Konfigurationsdatei



```
# a comment starts with '#' and will be ignored
# an include pattern starts with '+' and an exclude pattern with '-'
# every other line results in a warning
- ..* ..MonitoredClass.*(..)
+ private static void ..*.*(..)
- public java.util.List ..*.*(..)
+ public ..* ..*.*()
- public int evaluation.monitoredApplication.MonitoredClass.get*()
```

Gliederung

1. Einleitung
   - Motivation
   - Ansatz
   - Syntax
2. ProbeController
   - API
   - ProbeController
3. Konfiguration über Datei
   - Kieker Properties
   - Konfigurationsdatei
4. Live Demo