

# Logowanie zdarzeń w aplikacjach J2EE

[Marek.Berkan@e-point.pl](mailto:Marek.Berkan@e-point.pl)



4Developers, 4 kwietnia 2011

# Zakres wykładu

- **Cele logowania zdarzeń**
- **Sposób logowania**
- **Narzędzia**
- **Infrastruktura**
- **Ciekawe przypadki**

# Cele

- **Wsparcie w trakcie developmentu**
- **Diagnostowanie błędów w czasie testów**
- **Diagnostowanie błędów na produkcji**
- **Historia komunikacji z systemami zewnętrznymi**
- **Historia działań użytkowników**
- **Ochrona własnej reputacji**

# Wsparcie w trakcie developmentu

```
1  public void createOrder1() {
2      System.out.println("createOrder begin");
3
4      User user = usersM.getUser(login);
5      System.out.println("User found");
6
7      List<CartItem> items = cartM.getCartItems();
8      for(CartItem item : items) {
9          System.out.println("Add discount for item #" + item.getId());
10         item.setValue(item.getValue().multiply(user.getDiscount()));
11     }
12
13     System.out.println("XXXXXXXXXXXXXXXXXXXXXXXXXXXX before create order...");
14     System.out.println("=====");
15     ordersM.createOrder(user, items);
16     System.out.println("createOrder done");
17 }
```

# Diagnostowanie błędów w czasie testów

```
1 [3/30/11 11:08:14:053 GMT] UUID: B8HEFISIXORQ7XXQSNAJ ROOT: java.lang.RuntimeException: Babol swój widzę ogromny
2 -----
3 URL: http://de.wimp.v2.ext.e-point.pl/
4 Locale: en
5 Remote host: 194.50.49.34
6 Request Session ID: Y786Wic40Pvou2DFbA2N40_
7 Path info: /
8 Path translated: /opt/.../installedApps/aolweStgCell/wimp-de.ear/war-country-3.46.rc1.2.war
9 Query string: action=explode
10 Principal: null
11 Request parameters:
12     action: explode;
13 Headers:
14     Host: de.wimp.v2.ext.e-point.pl
15     User-Agent: Mozilla/5.0 (X11; U; Linux i686; pl-PL; rv:1.9.2.16) Gecko/20110323 Ubuntu/10.10
16 [...]
17 HttpSession:
18     TABLE_categories_tree: pl.epoint.ow.table.TableContext@6afa6afa[page=1,pageSize=-1,pagerSize=4,
19     pageSizes=[10, 20, 50, 100, 200],rootId=<null>,rootIdInInclusiveMode=false,expandedNodes=[]]
20     pl.epoint.foo.country.war.auth.ui.CountryAuthenticationConstants_login: test
21 -----
22 : pl.epoint.ow.services.processing.ProcessingException
23 Caused by: java.lang.reflect.InvocationTargetException
24     at java.lang.reflect.Method.invoke(Method.java:618)
25     at pl.epoint.ow.services.processing.impl.MethodDelegateAction.execute(MethodDelegateAction.java:59)
26     ... 64 more
27 Caused by: java.lang.RuntimeException: Babol swój widzę ogromny
28     at pl.epoint.foo.common.war.ExplodeActions.explodeRuntime(ExplodeActions.java:23)
29     ... 69 more
```

# Diagnostowanie błędów w czasie testów

## Application error

**Error has occured whilst performing this operation.**

If this problem is repeating itself, please contact Customer Services and give them the number presented below with information about the steps you performed that caused this error.

Sorry for any inconvenience caused.

Error number: **B8HEFISIX0RQ7XXQSNAJ**

[Start page](#)

# Diagnostowanie błędów w czasie testów

```
14 public String test1() {  
15     return numberFormat.format(ordersM.getLastOrderForUser(usersM.getUser(login)).getValue());  
16 }
```

```
1 Exception in thread "main" java.lang.NullPointerException  
2     at pl.epoint.samples.MultilineExecution.test1(MultilineExecution.java:15)  
3     at pl.epoint.samples.MultilineExecution.main(MultilineExecution.java:11)
```

# Diagnostowanie błędów w czasie testów

```
18 public String test2() {
19
20     User user = usersM.getUser(login);
21     assert user != null : "User with login '" + login + "' not found";
22
23     Order lastOrder = ordersM.getLastOrderForUser(user);
24     assert lastOrder != null : "User '" + login + "' has no order";
25
26     BigDecimal value = lastOrder.getValue();
27     assert value != null : "Order '" + lastOrder.getId() + "' has no value";
28
29     return numberFormat.format(value);
30 }
```

```
1 Exception in thread "main" java.lang.AssertionError: User 'abacki' has no order
2   at pl.epoint.samples.MultilineExecution.test2(MultilineExecution.java:24)
3   at pl.epoint.samples.MultilineExecution.main(MultilineExecution.java:11)
```



# Diagnostowanie błędów na produkcji

## Application error

**Error has occured whilst performing this operation.**

If this problem is repeating itself, please contact Customer Services and give them the number presented below with information about the steps you performed that caused this error.

Sorry for any inconvenience caused.

Error number: **B8HEFISIX0RQ7XXQSNAJ2011033011320201**

[Start page](#)

# Historia komunikacji z systemami zewnętrznymi

## Order processing history

Date and time	Action	Order status	Details
2011-03-02, 15:44:08	Order sent to B/E	Send to B/E for calculation	F/E order number: [REDACTED]
2011-03-02, 15:44:23	Update order by B/E	Temporarily confirmed	B/E order number: [REDACTED]
2011-03-02, 15:46:14	Send Saferpay SCD payment request	Temporarily confirmed	Process ID: '141761' CC refId: [REDACTED] CC name: 'Mastercard' Amount: '353.41 EUR'
2011-03-02, 15:46:16	Saferpay SCD payment successful	Accepted by user/Payment made	Process ID: '141761' Response ID: [REDACTED]
2011-03-02, 15:46:22	Update order status by B/E	Confirmed	
2011-03-03, 01:12:57	Update order status by B/E	Ready for shipment	New package number: , New delivery tracking link:
2011-03-03, 14:17:55	Update order status by B/E	Shipped	New package number: [REDACTED], New delivery tracking link: <a href="http://www.dpd.de">http://www.dpd.de</a>

# Historia działań użytkowników

## Event list

<u>Event type</u>	<u>Event group</u>	<u>Item name</u>	<u>Item id</u>	<u>Date and time (1) ▲</u>	<u>User</u>	<u>Priority</u>
<u>Editing language version content</u>	CMS	bloom	11474	2011-02-28 14:59	operator (John Peltola)	Low
<u>Adding language version</u>	CMS	bloom	11474	2011-02-28 14:58	operator (John Peltola)	Low
<u>Adding document</u>	CMS	bloom	11474	2011-02-28 14:58	operator (John Peltola)	Low
<u>Editing language version content</u>	CMS	bloom	11462	2011-02-28 14:57	operator (John Peltola)	Low
<u>Editing language version content</u>	CMS	HR_RecruitmentSpecialist	11473	2011-02-28 14:56	operator (John Peltola)	Low
<u>Editing RMO teaser</u>	RMOnline	LOM 4, LOM 4	10229	2011-02-28 14:56	operator (John Peltola)	Low
<u>Editing RMO teaser</u>	RMOnline	LOM 4, LOM 4	10230	2011-02-28 14:53	operator (John Peltola)	Low
<u>Login as user</u>	User accounts	9320	14594	2011-02-28 14:52	operator (John Peltola)	Medium
<u>Administrator</u>	Administrator	Administrator	10194	2011-02-28 14:52	operator (John Peltola)	Low

# Sposób logowania

- `System.out.println`
- **Kod diagnostyczny ogólny**
- **Kod diagnostyczny zorientowany na aspekty**
- **Poziomy logowania (debug, info, warn, error, itp.)**
- **Logowanie zdarzeń do bazy**

# commons-logging

<http://commons.apache.org/logging/> 2002-2007

- **Fasada a nie konkretna implementacja**
- **Dla bibliotek a nie konkretnych instalacji**
- **Automatyczne poszukiwanie implementacji**
- **Problemy z ClassLoader'ami**

# commons-logging

```
1 package pl.epoint.commonslogging;
2
3 import org.apache.commons.logging.Log;
4 import org.apache.commons.logging.LogFactory;
5
6 public class CommonsLoggingTest {
7
8     private static final Log log = LogFactory.getLog(CommonsLoggingTest.class);
9
10    public static void main(String[] args) {
11        log.debug("Hello world!");
12        log.error("My error", new RuntimeException());
13    }
14 }
```

```
1 log4j.rootLogger=DEBUG, A1
2
3 log4j.appender.A1=org.apache.log4j.ConsoleAppender
4 log4j.appender.A1.layout=org.apache.log4j.PatternLayout
5 log4j.appender.A1.layout.ConversionPattern=%d %5p (%C:%M:%L) - %m%n
```

```
1 2011-03-29 15:26:14,110 DEBUG (pl.epoint.commonslogging.CommonsLoggingTest:main:11) - Hello world!
2 2011-03-29 15:26:14,113 ERROR (pl.epoint.commonslogging.CommonsLoggingTest:main:12) - My error
3 java.lang.RuntimeException
4     at pl.epoint.commonslogging.CommonsLoggingTest.main(CommonsLoggingTest.java:12)
```

# log4j

<http://logging.apache.org/log4j/> Ceki Gülcü, 1999-2007

- **Konkretna implementacja**
- **Najpopularniejsza**
- **Pełna dokumentacja płatna ok. 15 USD**
- **MDC (*ang. Mapped Diagnostic Context*)**

# log4j

```
1 package pl.epoint.log4j;
2
3 import org.apache.log4j.Logger;
4 import org.apache.log4j.MDC;
5
6 public class Log4jTest {
7
8     private static final Logger log = Logger.getLogger(Log4jTest.class);
9
10    public static void main(String[] args) {
11        MDC.put("login", "abacki");
12        log.debug("Hello world!");
13        log.error("My error", new RuntimeException());
14    }
15 }
```

```
1 log4j.rootLogger=DEBUG, A1
2
3 log4j.appender.A1=org.apache.log4j.ConsoleAppender
4 log4j.appender.A1.layout=org.apache.log4j.PatternLayout
5 log4j.appender.A1.layout.ConversionPattern=%d %5p (%C:%M:%L) - %m (login: %X{login})%n
```

```
1 2011-03-29 15:37:25,021 DEBUG (pl.epoint.log4j.Log4jTest:main:12) - Hello world! (login: abacki)
2 2011-03-29 15:37:25,024 ERROR (pl.epoint.log4j.Log4jTest:main:13) - My error (login: abacki)
3 java.lang.RuntimeException
4     at pl.epoint.log4j.Log4jTest.main(Log4jTest.java:13)
```



# java.util.logging

`http://download.oracle.com/javase/1.4.2/docs/guide/util/logging/overview.html` od 2002

- **Konkretna implementacja**
- **Zawarta w JDK od wersji 1.4**

# java.util.logging

```
1 package pl.epoint.jdk;
2
3 import java.util.logging.Level;
4 import java.util.logging.Logger;
5
6 public class JdkTest {
7
8     private static final Logger log = Logger.getLogger(JdkTest.class.toString());
9
10    public static void main(String[] args) {
11        log.fine("Hello world!");
12        log.log(Level.SEVERE, "My error", new RuntimeException());
13    }
14 }
```

```
1 handlers=java.util.logging.ConsoleHandler
2 .level=FINE
3
4 java.util.logging.ConsoleHandler.level=ALL
```

```
1 2011-03-29 15:52:55 pl.epoint.jdk.JdkTest main
2 FINE: Hello world!
3 2011-03-29 15:52:55 pl.epoint.jdk.JdkTest main
4 SEVERE: My error
5 java.lang.RuntimeException
6     at pl.epoint.jdk.JdkTest.main(JdkTest.java:12)
```

# SLF4J

<http://slf4j.org/> Ceki Gülcü, od 2005

- **Fasada a nie konkretna implementacja**
- **Ulepszona wersja commons-logging**
- **MDC (*ang. Mapped Diagnostic Context*)**

# SLF4J + log4j

```
1 package pl.epoint.slf4j;
2
3 import org.slf4j.*;
4
5 public class Sslf4jTest {
6
7     private static final Logger log = LoggerFactory.getLogger(Sslf4jTest.class);
8
9     public static void main(String[] args) {
10         MDC.put("login", "abacki");
11         log.debug("Execute: " + new Test() + " result"); // slow
12         log.info("Execute: {} {}", new Test(), "result"); // fast
13     }
14 }
```

```
1 log4j.rootLogger=INFO, A1
2
3 log4j.appender.A1=org.apache.log4j.ConsoleAppender
4 log4j.appender.A1.layout=org.apache.log4j.PatternLayout
5 log4j.appender.A1.layout.ConversionPattern=%d %5p (%C:%M:%L) - %m (login: %X{login})%n
```

```
1 2011-03-29 16:19:21,500 WARN (pl.epoint.slf4j.Sslf4jTest$Test:toString:17) - Test.toString (login: abacki)
2 2011-03-29 16:19:21,503 WARN (pl.epoint.slf4j.Sslf4jTest:main:12) - Test.toString (login: abacki)
3 2011-03-29 16:19:21,503 INFO (pl.epoint.slf4j.Sslf4jTest:main:12) - Execute: something result (login: abacki)
```

# logback

<http://logback.qos.ch/> Ceci Gülcü, od 2006

- **Konkretna implementacja**
- **Ulepszona wersja log4j tego samego autora**
- **Brak stabilnej wersji (0.9.28)**
- **Wybrane ulepszenia:**
  - 10 krotnie szybsza implementacja
  - pełne pokrycie testami
  - natywna integracja z SLF4J
  - automatyczne odświeżanie konfiguracji
  - pełna dokumentacja
  - wiele JVM-ów może logować do tego samego pliku
  - filtry

# logback

```
1 package pl.epoint.logback;
2 import org.slf4j.*;
3
4 public class LogbackTest {
5     private static final Logger log = LoggerFactory.getLogger(LogbackTest.class);
6     public static void main(String[] args) {
7         MDC.put("login", "abacki");
8         log.info("Execute: {} {}", new Test(), "result");
9         log.error("My error", new RuntimeException());
10    }
11 }
```

```
1 <configuration>
2   <appender name="STDOUT" class="ch.qos.logback.core.ConsoleAppender">
3     <encoder>
4       <pattern>%d{HH:mm:ss.SSS} [%thread] %-5level %logger{36}:%L - %msg \(\login: %X{login}\) %n</pattern>
5     </encoder>
6   </appender>
7   <root level="INFO">
8     <appender-ref ref="STDOUT" />
9   </root>
10 </configuration>
```

```
1 16:32:25.194 [main] WARN pl.epoint.logback.LogbackTest:14 - Test.toString (login: abacki)
2 16:32:25.196 [main] INFO pl.epoint.logback.LogbackTest:8 - Execute: something result (login: abacki)
3 16:32:25.200 [main] ERROR pl.epoint.logback.LogbackTest:9 - My error (login: abacki)
4 java.lang.RuntimeException: null
5 at pl.epoint.logback.LogbackTest.main(LogbackTest.java:9) [classes/:na]
```

# Instrastruktura

- **Ewidencja plików logów:**
  - Nazwa, opis
  - Zarządca (log4j, syslog, skrypty)
  - Polityka starzenia (rotacja, gzipowanie, usuwanie)
  - Szacowana wielkość
- **Agregacja**

# Instrastruktura

Nazwa, opis	Serwery	Plik	Rotate	Czyszczenie	Kopia	Rozmiar	Status
<b>SystemOut</b> - wspólne logi na System.out (WAS)	was10-17	/opt/WebSphereV61/AppServer /profiles/aol{ee,ru} /logs/{Prod,ProdRu}Cluster-was?-? /SystemOut.log	północ przez WAS	po 10 dniach logi przenoszone są do /old, gzipowane, a po dodatkowych 80 dniach usuwane (log-admin.sh)	TAK	6,4 MB	✓
<b>SystemErr</b> - wspólne logi na System.err (WAS)	was10-17	/opt/WebSphereV61/AppServer /profiles/aol{ee,ru} /logs/{Prod,ProdRu}Cluster-was?-? /SystemErr.log	północ przez WAS	po 10 dniach logi przenoszone są do /old, gzipowane, a po dodatkowych 80 dniach usuwane (log-admin.sh)	TAK	0,2 MB	✓
<b>native_stdout.log</b> - logi z JVM WASa	was10-17	/opt/WebSphereV61/AppServer /profiles/aol{ee,ru} /logs/{Prod,ProdRu}Cluster-was?-? /native_stdout.log	logrotate, dobowe	po 90 rotowaniach logi są usuwane		0	✓
<b>native_stderr.log</b> - logi z JVM WASa	was10-17	/opt/WebSphereV61/AppServer /profiles/aol{ee,ru} /logs/{Prod,ProdRu}Cluster-was?-? /native_stderr.log	logrotate, dobowe	po 90 rotowaniach logi są usuwane		1,72 MB	✓
<b>aplikacja j2ee</b> - log4j z aplikacji EE, RU	was10-17	/opt/WebSphereV61/AppServer /profiles/aol/logs /{Prod,ProdRu}Cluster- was?-server?/vebso.log	logrotate, dobowe	po 90 rotowaniach logi są usuwane	TAK	18 MB	✓
<b>crontasc</b> s - logi z wywołań skryptów perlowych	wszystkie	/var/log/vebso /vebso-{PR,PRRU}_yyyy-MM-dd.log	logrotate, miesięcznie	po 12 rotowaniach logi są usuwane	TAK	różnie, ale mało	✓



# Automatyczne przeglądy logów

```
1 Wyjatkami:
2   984 java.lang.NullPointerException
3     27 javax.transaction.RollbackException: Transaction rolled back
4     21 com.ibm.wsspi.sib.core.exception.SIConnectionDroppedException: CWSIJ0047E: An operation was attempted
5       on a connection that is already closed.
6     20 ] CWSJR1102E: An internal error has occurred. The exception javax.resource.ResourceException: enlist:
7       caught Exception was thrown
8     15 pl.epoint.ow.services.processing.ProcessingException: Process name is undefined (has null value).
9     9 org.apache.velocity.exception.ResourceNotFoundException: Unable to find resource
10    'catalog/resource_not_found.vt'
11    7 pl.epoint.ow.services.processing.ProcessingException: Form name parameter does not exist.
12
13 NullPointery:
14   894 at pl.epoint.foo.common.war.util.PathUtils.splitPath(PathUtils.java:33)
15    84 at net.sf.ehcache.Cache.isExpired(Cache.java:1043)
16    6 at pl.epoint.foo.country.or.OrderStatusDAOImpl.getOrderStatus(OrderStatusDAOImpl.java:59)
17
18 Asercje:
19   15 AssertionError
20   14 Model is null - user logged out?
21   12 foo representative is null - logged out?
22    4 orderIdAsInteger is null
23    4 Current registered foo representative is null. (AOD-2788)
24    2 User is not an registered user.
25
26 Statystyka wystapien znanych wyjatkov:
27   AOD-4374 0
28   AOD-5261 12031
29   AOD-6136 7
```

# Ciekawe przypadki (1)

- „Chcieliśmy otrzymywać informacje o każdym błędzie e-mailem...”
- Nie logujemy wszystkiego...

## Ciekawe przypadki (2)

- „Policja żąda IP użytkownika który złożył to zamówienie”
- „Usunęliśmy dokument a on znowu jest widoczny”

# Podsumowanie

## Podsumowanie

# Czy są jakieś pytania?

## Zapraszam do dyskusji