



码出高效

阿里巴巴Java开发手册 终极版 (1.3.0)

Alibaba Java Coding Guidelines

前言

1.3.0

前言

一、编程规约	1
(一) 命名风格	1
(二) 常量定义	3
(三) 代码格式	4
(四) OOP 规约	6
(五) 集合处理	9
(六) 并发处理	12
(七) 控制语句	14
(八) 注释规约	16
(九) 其它	17
二、异常日志	18
(一) 异常处理	18
(二) 日志规约	19
三、单元测试	21
四、安全规约	23
五、MySQL 数据库	24
(一) 建表规约	24
(二) 索引规约	25
(三) SQL 语句	27
(四) ORM 映射	28
六、工程结构	30
(一) 应用分层	30
(二) 二方库依赖	31
(三) 服务器	32
附 1：版本历史	34
附 2：本手册专有名词	35

(注：浏览时请使用 PDF 左侧导航栏)

Java

1.3.0		2017.9.25	PDF

一、编程规约

(一) 命名风格

1.

`_name / __name / $Object / name_ / name$ / Object$`

2.

`alibaba / taobao / youku / hangzhou`
`DaZhePromotion [] / getPingfenByName() [] / int = 3`

3.

`UpperCamelCase` `DO / BO /`
`DTO / VO / AO`
`MarcoPolo / UserDO / XmlService / TcpUdpDeal / TaPromotion`
`macroPolo / UserDo / XMLService / TCPUDPDeal / TAPromotion`

4.

`lowerCamelCase`
`localValue / getHttpMessage() / inputUserId`

5.

`MAX_STOCK_COUNT`
`MAX_COUNT`

6.

`Abstract Base Exception`
`Test`

7.

`String[] args;`
`String args[]`

8.

`POJO` `is`
`Boolean isDeleted` `isDeleted() RPC`

“ ” deleted

9.

com.alibaba.open.util MessageUtils
spring

10.

AbstractClass AbsClass condition condi

11.

PullCodeFromRemoteRepository
int a;

12.

public class OrderFactory;
public class LoginProxy;
public class ResourceObserver;

13.

public
Javadoc
void f();
String COMPANY = "alibaba";
public abstract void f();
JDK8 default

14.

1 Service DAO SOA
Impl
CacheServiceImpl CacheService
2 -able
AbstractTranslator Translatable

15.

Enum

ProcessStatusEnum

SUCCESS / UNKOWN_REASON

16.

A) Service/DAO

- 1 get
- 2 list
- 3 count
- 4 save/insert
- 5 remove/delete
- 6 update

B)

- 1 xxxDO xxx
- 2 xxxDTO xxx
- 3 xxxVO xxx
- 4 POJO DO/DTO/BO/VO xxxPOJO

(二) 常量定义

1.

```
String key = "Id#taobao_" + tradeId;
cache.put(key, value);
```

2.

```
long Long L l 1

Long a = 2l; 21 Long 2?
```

3.

CacheConsts

ConfigConsts

4.

- 1 client.jar constant
- 2 modules constant

“ ”

```
A    public static final String YES = "yes";
B    public static final String YES = "y";
A.YES.equals(B.YES)    true    false
```

```
3    constant
4    constant
5    private static final
```

5.

```
public Enum { MONDAY(1), TUESDAY(2), WEDNESDAY(3), THURSDAY(4), FRIDAY(5), SATURDAY(6),
SUNDAY(7);}
```

(三) 代码格式

1. { }

```
1
2
3
4    else
```

2.

```
5
if (      a == b      )
```

3. if/for/while/switch/do

4.

=

5. 4 tab

tab 1 tab 4 IDEA tab 4

Use tab character eclipse insert spaces for tabs

1-5

```
public static void main(String[] args) {
    //    4
    String say = "hello";
    //
    int flag = 0;
```


10.

```
int a = 3;
long b = 4L;
float c = 5F;
StringBuffer sb = new StringBuffer();

sb          a  b  c
```

11.

(四) OOP 规约

1.

2. Override

getObject()	get0bject()	0	0	Override
-------------	-------------	---	---	----------

3. Java Object

```
public User getUsers(String type, Integer... ids) {...}
```

4.

Deprecated

5.

```
java.net.URLDecoder          decode(String encodeStr)
                             decode(String source, String encode)
```

6. Object equals

equals

```
"test".equals(object);
object.equals("test");

java.util.Objects#equals  JDK7
```

7. equals

Integer var = ?	-128 127	Integer
-----------------	-------------	---------

IntegerCache.cache

Integer

==

equals

8.

1 POJO

2 RPC

3

POJO

NPE

null

NPE

x% x

RPC

0%

null

9.

DO/DTO/VO POJO

POJO gmtCreate new Date();

10.

serialVersionUID

serialVersionUID

serialVersionUID

11.

init

12.

POJO toString

IDE

source> generate toString

POJO

super.toString

POJO toString()

13.

String split

IndexOutOfBoundsException

```
String str = "a,b,c,, ";
String[] ary = str.split(",");
//      3      3
System.out.println(ary.length);
```

14.

15

15.

>

> getter/setter

“ ”

Service DAO getter/setter

16. setter this. =
getter/setter

```
public Integer getData() {
    if (true) {
        return this.data + 100;
    } else {
        return this.data - 100;
    }
}
```

17. StringBuilder append
new StringBuilder
append toString String

```
String str = "start";
for (int i = 0; i < 100; i++) {
    str = str + "hello";
}
```

18. final final
1 String
2 POJO
3 POJO setter
4
5 final

19. Object clone
clone clone

20.
1 new private
2 public default
3 static protected
4 static private
5 static private

6	static	final
7		private
8		protected

	private	public	service
public			

(五) 集合处理

- hashCode equals
 - equals hashCode
 - Set hashCode equals Set
 - Map hashCode equals
 - String hashCode equals String
key
- ArrayList subList ArrayList ClassCastException
java.util.RandomAccessSubList cannot be cast to java.util.ArrayList.
 - subList ArrayList SubList ArrayList
 - ArrayList SubList
- subList
ConcurrentModificationException
- toArray(T[] array)
 - list.size()
 - toArray toArray
[list.size()]
 - null

```
List<String> list = new ArrayList<String>(2);
list.add("guan");
list.add("bao");
String[] array = new String[list.size()];
array = list.toArray(array);
```

toArray Object[]

ClassCastException

5. Arrays.asList()

add/remove/clear UnsupportedOperationException

asList Arrays Arrays.asList

```
String[] str = new String[] { "you", "wu" };
```

```
List list = Arrays.asList(str);
```

```
list.add("yangguanbao");
```

```
str[0] = "gujin"; list.get(0)
```

6. <? extends T> add

<? super T> get

PECS(Producer Extends Consumer Super)

<? extends T> <? super T>

7. foreach remove/add remove Iterator

Iterator

```
Iterator<String> iterator = list.iterator();
```

```
while (iterator.hasNext()) {
```

```
String item = iterator.next();
```

```
if ( ) {
```

```
iterator.remove();
```

```
}
```

```
}
```

```
List<String> list = new ArrayList<String>();
```

```
list.add("1");
```

```
list.add("2");
```

```
for (String item : list) {
```

```
if ("1".equals(item)) {
```

```
list.remove(item);
```

```
}
```

```
}
```

"1" "2"

8. JDK7 Comparator Arrays.sort

Collections.sort IllegalArgumentException

1 x y y x

2 x>y y>z x>z
 3 x=y x z y z

```
new Comparator<Student>() {
    Override
    public int compare(Student o1, Student o2) {
        return o1.getId() > o2.getId() ? 1 : -1;
    }
};
```

9.

```
HashMap        HashMap(int initialCapacity)
initialCapacity = (                                /                                ) + 1                                loader
factor                                0.75                                16
HashMap                                1024
7                                resize                                hash
```

10.

```
entrySet        Map        KV        keySet
keySet                                2                                Iterator                                hashMap
key        value        entrySet                                key        value        entry
JDK8        Map.foreach
values()                                V                                list                                keySet()        K
Set                                entrySet()                                K-V
```

11.

Map K/V null

| | Key | Value | Super | |
|-------------------|------|-------|-------------|----------|
| Hashtable | null | null | Dictionary | |
| ConcurrentHashMap | null | null | AbstractMap | JDK8:CAS |
| TreeMap | null | null | AbstractMap | |
| HashMap | null | null | AbstractMap | |

HashMap ConcurrentHashMap null

null NPE

12.

(sort) (order) (unsort)

(unorder)

ArrayList order/unsort HashMap unorder/unsort TreeSet

order/sort

| | | |
|-----|----------|------|
| 13. | Set | List |
| | contains | |

(六) 并发处理

1.

2.

```
public class TimerTaskThread extends Thread {
    public TimerTaskThread() {
        super.setName("TimerTaskThread");
        ...
    }
}
```

3.

“ ”

4. Executors ThreadPoolExecutor

Executors

1 FixedThreadPool SingleThreadPool:

Integer.MAX_VALUE OOM

2 CachedThreadPool ScheduledThreadPool:

Integer.MAX_VALUE OOM

5. SimpleDateFormat static

static DateUtils

DateUtils

```
private static final ThreadLocal<DateFormat> df = new ThreadLocal<DateFormat>() {
    Override
    protected DateFormat initialValue() {
        return new SimpleDateFormat("yyyy-MM-dd");
    }
};
```

JDK8

Instant

Date

LocalDateTime

Calendar

DateTimeFormatter SimpleDateFormat

simple beautiful strong

immutable thread-safe

6.

RPC

7.

A B C
A B C

8.

version
20%
3

9.

Timer TimeTask
ScheduledExecutorService

10.

CountDownLatch countDown
catch countDown
await
try-catch

11.

Random
seed
Random java.util.Random Math.random()
JDK7 API ThreadLocalRandom JDK7

12.

double-checked locking
(The "Double-Checked Locking is Broken" Declaration)
JDK5 volatile

```
class Singleton {
    private Helper helper = null;
    public Helper getHelper() {
        if (helper == null) synchronized(this) {
            if (helper == null)
                helper = new Helper();
        }
        return helper;
    }
    // other methods and fields...
}
```


1. switch case break/return
case switch default
2. if/else/for/while/do
if (condition) statements;
3. if-else
if (condition) {
...
return obj;
}
// else ;
if()...else if()...else...
3
3 if-else

public void today() {
if (isBusy()) {
System.out.println("change time.");
return;
}

if (isFree()) {
System.out.println("go to travel.");
return;
}
}

```
        System.out.println("stay at home to learn Alibaba Java Coding Guidelines.");
        return;
    }
}
```

4. getXxx/isXxx

if

```
//
final boolean existed = (file.open(fileName, "w") != null)    (...) || (...);
if (existed) {
    ...
}

if ((file.open(fileName, "w") != null)    (...) || (...)) {
    ...
}
```

5.

try-catch

try-catch

6.

7.

1

2

3

4

RPC/API/HTTP

5

8.

1

2

DAO

Service

DAO

3

private

(八) 注释规约

1. Javadoc `/** */`
`// xxx`
 IDE Javadoc Javadoc
 IDE

2. Javadoc

3.

4. `//`
`/* */`

5.

6. “ ”
 “TCP” “ ”

7.

8.

1 2

9.

10.

`// put elephant into fridge`
`put(elephant, fridge);`

put elephant fridge

11.

```
1      TODO : [ ]
      Javadoc Javadoc
      Javadoc
2      FIXME : [ ]
      FIXME
```

(九) 其它

1.

```
Pattern pattern = Pattern.compile( );
```

2.

```
velocity POJO
POJO getXxx() boolean boolean is
isXxx()
Boolean getXxx()
```

3.

```
!{var}—
var=null {var}
```

4.

```
Math.random() double 0≤x<1
x 10
Random nextInt nextLong
```

5.

```
System.currentTimeMillis(); new Date().getTime();
System.nanoTime() JDK8
Instant
```

6.

MVC

7.

8.

(///)

二、异常日志

(一) 异常处理

- Java RuntimeException

catch IndexOutOfBoundsException NullPointerException

catch

NumberFormatException

if (obj != null) {...}

try { obj.method() } catch (NullPointerException e) {...}
-
- try-catch catch

catch
-
- try catch
- finally try-catch

JDK7 try-with-resources
- finally return finally return

try return
-
- null
null null NPE
NPE

null
- null
- NPE NPE

1 return NPE

public int f() { return Integer } null NPE

| | | |
|-----|---|---------------------------|
| 2 | null | |
| 3 | isEmpty | null |
| 4 | | NPE |
| 5 | Session | NPE |
| 6 | obj.getA().getB().getC()
JDK8 Optional | NPE |
| 11. | unchecked / checked
Exception Throwable
DAOException / ServiceException | new RuntimeException() |
| 12. | “ ” “ ”
“ ”
isSuccess() “ ” “ ”
RPC Result | http/api
RPC
Result |
| 1 | | |
| 2 | new | error message |
| 13. | Don't Repeat Yourself | DRY |
| | public
private boolean checkParam(DTO dto) {...} | |

(二) 日志规约

| | | |
|----|---|-----|
| 1. | Log4j Logback | API |
| | SLF4J API | |
| | import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
private static final Logger logger = LoggerFactory.getLogger(ABC.class); | |
| 2. | 15 | “ ” |
| 3. | appName_logType_logName.log logType:
stats/desc/monitor/visit logName: | |

mppserver
mppserver_monitor_timeZoneConvert.log

4. trace/debug/info

```
logger.debug("Processing trade with id: " + id + " and symbol: " + symbol);
warn symbol
toString()

if (logger.isDebugEnabled()) {
    logger.debug("Processing trade with id: " + id + " and symbol: " + symbol);
}

logger.debug("Processing trade with id: {} and symbol : {} ", id, symbol);
```

5. log4j.xml additivity=false

```
<logger name="com.taobao.dubbo.config" additivity="false">
```

6.

```
throws
logger.error(toString + "_" + e.getMessage(), e);
```

7. debug info

warn

8. warn

```
error
error
```


9.

BCDE

- B Border
- C Correct
- D Design
- E Error

10.

11.

RDC

RDC_UNIT_TEST_

12.

13.

UC

14.

15.

-
-
-
-

16.

-
-
-
-

四、安全规约

1.

2.

:158****9119 4

3.

SQL METADATA SQL
SQL

4.

- page size
 - order by
 -
 - SQL
 -
 - ReDoS
- Java

5.

HTML

6.

AJAX CSRF

CSRF(Cross-site request forgery)

CSRF / URL

7.

8.

五、MySQL 数据库

(一) 建表规约

1. `is_xxx` `unsigned tinyint`

`1` `0`

`unsigned`

`is_deleted` `1` `0`
- MySQL Windows Linux

`aliyun_admin rdc_config level3_name`

`AliyunAdmin rdcConfig level_3_name`
3. `DO`
4. `desc range match delayed` MySQL
5. `pk_` `uk_` `idx_`

`pk_` `primary key` `uk_` `unique key` `idx_` `index`
6. `decimal` `float` `double`

`float` `double`

`decimal`
7. `char`
8. `varchar` `5000`

`text`
9. `id, gmt_create, gmt_modified`

`id` `unsigned bigint` `1 gmt_create,`

`gmt_modified` `date_time`
10. `" _ "`

`alipay_task / force_project / trade_config`

11.
12.
13.

1

2

varchar

text
14.

500

2GB
15.

| | 150 | unsigned tinyint | 1 | 0 255 |
|--|-----|-------------------|---|---------|
| | | unsigned smallint | 2 | 0 65535 |
| | | unsigned int | 4 | 0 42.9 |
| | 50 | unsigned bigint | 8 | 0 10 19 |

(二) 索引规约

1.

insert
2.

join

join

join

SQL
3.

varchar

90% count(distinct left(,))/count(*)

4.

B-Tree

5.

order by

order by

file_sort

where a=? and b=? order by c;

a_b_c

WHERE a>10 ORDER BY b;

a_b

6.

11

11

explain

extra

using index

7.

MySQL

offset

offset+N

offset

N

offset

SQL

id

SELECT a.* FROM 1 a, (select id from 1 where LIMIT 100000,20) b where a.id=b.id

8.

SQL

range

ref

consts

1 consts

2 ref

normal index

3 range

explain

type=index

index

range

9.

where a=? and b=? a

idx_a

where a>?

and b=?

a

b

10.

11.

1

2

3

“ ”

(三) SQL 语句

1. count() count() count(*) count(*) SQL92

NULL NULL

count(*) NULL count() NULL

2. count(distinct col) NULL count(distinct

col1, col2) NULL 0

3. NULL count(col) 0 sum(col)

NULL sum() NPE

sum NPE SELECT IF(ISNULL(SUM(g)),0,SUM(g))

FROM table;

4. ISNULL() NULL

NULL NULL

1 NULL<>NULL NULL false

2 NULL=NULL NULL true

3 NULL<>1 NULL true

5. count 0

6.

student_id

student_id

student_id

student_id

7.

8. select

```

1.          *
          1          2          resultMap
2.          POJO          is          is_          resultMap
          POJO          <resultMap>
MyBatis Generator
3.          resultClass
          DO
4.          sql.xml          #{ } #param#          ${ }          SQL
5.          iBATIS          queryForList(String statementName,int start,int size)
          statementName          SQL          subList
start,size
Map<String, Object> map = new HashMap<String, Object>();
map.put("start", start);
map.put("size", size);
6.          HashMap  Hashtable
resultClass="Hashtable"

```

7. gmt_modified
8. POJO

update table set c1=value1,c2=value2,c3=value3;

SQL

binlog
9. QPS

Transactional
10. compareValue

<isEqual>

null

<isNotNull>

null

<isEmpty>

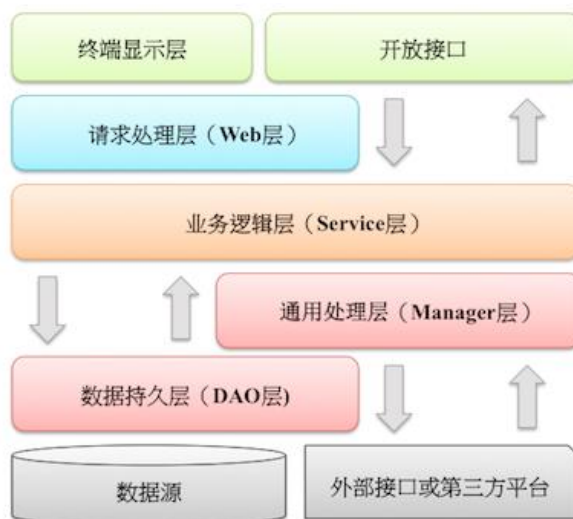
六、工程结构

(一) 应用分层

1.

Web

Service



- Service RPC Web http
- velocity JS
- JSP
- Web
- Service
- Manager
 - 1
 - 2 Service
 - 3 DAO DAO
- DAO MySQL Oracle Hbase
- RPC HTTP

2.

DAO

catch catch(Exception e) throw new DAOException(e)

Manager/Service

Service

Manager

Service

DAO

Service

Web

3.

- **DO** Data Object DAO
- **DTO** Data Transfer Object Service Manager
- **BO** Business Object Service
- **AO** Application Object Web Service
- **VO** View Object Web
- **Query** 2
Map

(二) 二方库依赖

1. GAV

- 1 **GroupID** com.{ /BU }. .[] 4
{ /BU} alibaba/taobao/tmall/aliexpress BU
com.taobao.jstorm com.alibaba.dubbo.register
- 2 **ArtifactID** -
dubbo-client / fastjson-api / jstorm-tool
- 3 **Version**

2.

- 1 API
- 2 API
- 3 BUG
1.0.0 0.0.1
1.3.3
1.3.4 1.4.0 2.0.0

3.

SNAPSHOT
SNAPSHOT

4.

dependency:resolve

dependency:tree
 <excludes>
 jar
5.

POJO
6.

springframework-core,-context,-beans

\${spring.version}
7.

pom

GroupId
 ArtifactId

 Version

war

 lib
8.

pom
 <dependencies>
 <dependencyManagement>
 <dependencyManagement>
 version
 scope
 pom
 <dependencies>
 pom
 <dependencies>
- 9.
10.

1

API
 Service API

 Utils
 provided
 log
 2

(三) 服务器

1.

TCP
 time_wait
 240
 time_wait
 time_wait
- linux
 /etc/sysctl.conf
 net.ipv4.tcp_fin_timeout = 30

2.

| | | | |
|--|----|-----------------------|---------|
| | | File Descriptor | fd |
| | | TCP/UDP | |
| | fd | linux | fd 1024 |
| | fd | "open too many files" | linux |
3.

| | | | | |
|------|-----|---------------------------------|-----|-----|
| | JVM | -XX:+HeapDumpOnOutOfMemoryError | JVM | OOM |
| dump | | | | |
| OOM | | | | |
4.

| | | | | |
|--|-----|-----|-----|----|
| | JVM | Xms | Xmx | GC |
|--|-----|-----|-----|----|
5.

| | | | |
|-----|--|---------|-----|
| | | forward | URL |
| URL | | | |

34 / 35

附 2：本手册专有名词

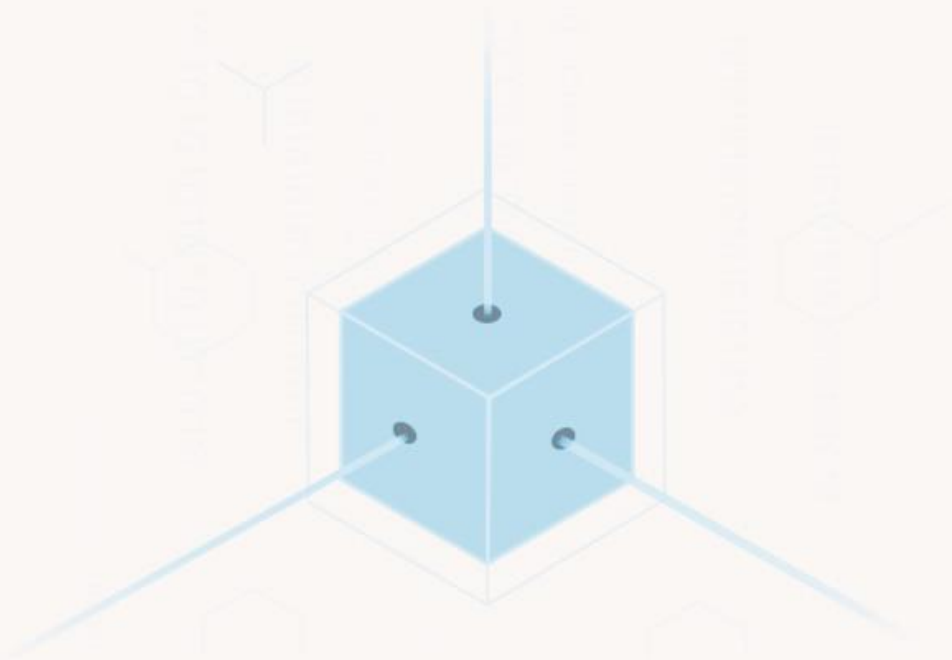
1. **POJO** Plain Ordinary Java Object : POJO setter / getter
/ toString DO/DTO/BO/VO

2. **GAV** GroupId ArtifactId Version : Maven jar

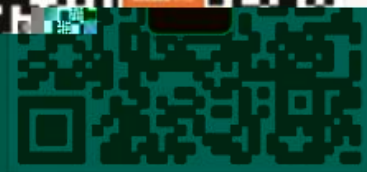
4. **ORM** Object Relation Mapping :

5. **NPE** java.lang.NullPointerException :

~~6. **SOA** Service Oriented Architecture :~~



关注阿里技术



访问阿里云栖社区

