

# read resource/property file

websecurity.database.url

private static Properties mProps = null;

final URL propsUrl = this.getClass().getClassLoader().getResource(APP\_PROPERTIES\_FILE);

InputStream stream = null;

stream = propsUrl.openStream();

mProps.load(stream);

stream.close();

i18n support (select file based on local)

java.util.ResourceBundle r= java.util.ResourceBundle.getBundle("data/java-system-properties.txt");

System.out.println(r.getKeys());

*java.util.MissingResourceException:*

*Can't find bundle for base name data/java-system-propertiess.txt, locale en\_US*

# stringman/format

new String().starts With("SELECT")

……………………………………………………………………………………………………………………………………………………………

……………………………………………………………………………………………………………………………………………………………

String.format ("Field [%s] needs to be a primitive; found [%s]",

fieldName, foi.getTypeName()));

also usable as System.out.format()

|  |  |  |  |
| --- | --- | --- | --- |
| %d=integer as is  %6d=padding left to 6  %-6d=padding right to 6  %06d=padding left to 6 use 0 for padding    %.2d=trim to 2 digits | %s=string as is  %5s=padding left to 5  %-5s=padding right to 5  %.8d=trim to 8 chars | %f=float as is  %9.4f=trimt digits to 4, padd until 9, align to left | *%n=new line* |

full specs: http://docs.oracle.com/javase/6/docs/api/java/util/Formatter.html#syntax

……………………………………………………………………………………………………………………………………………………………

String.format("%20s","").replace("<space> ","-");

String.format("<%s>", "abcc ".trim());

<abcc>

……………………………………………………………………………………………………………………………………………………………

StringTokenizer st = new StringTokenizer(toParse,DELIMITER);

st.nextToken();

return st.nextToken();

……………………………………………………………………………………………………………………………………………………………

"abc.aaa.".indexOf(".")

"abc.aaa.".lastIndexOf(".")

"abc".startsWith("a")

"abc".endsWith("c")

"abcccd".replaceAll("c", "x")

String.valueOf(3)

……………………………………………………………………………………………………………………………………………………………

"10.11.2013".matches("^\\d+\\.\\d+\\.\\d+")

……………………………………………………………………………………………………………………………………………………………

**org.apache.commons.lang.StringUtils**

StringUtils.defaultIfEmpty ()

StringUtils.lowerCase()

StringUtils.upperCase()

**org.apache.commons.lang.text.StrLookup;**

**org.apache.commons.lang.text.StrSubstitutor;**

emailMsg.setBody(new StrSubstitutor(new StrLookup() {

@Override

public String lookup(final String key) {

return NotifGtwUtil.getPayloadFieldMandatory(notificationMessage, key);

}

}).replace(bodyTemplate));

# conversion

Note: Arrays and Collections methods to manipulate arrays/collections

**Character**.*toUpperCase*(“abc”.charAt(1)) //B

BigDecimal to String/Rounding-via-string-format

BigDecimal less=new BigDecimal("10.2345");

System.out.format("%.2f", less.floatValue()); //print 10.23

BigDecimal to nice String

// Locale.ITALIAN -1.196,53 Locale.US -1,196.53

System.out.format (“%s”, java.text.NumberFormat.getInstance(Locale.ITALIAN).format(sum));

Print an array

**System**.*out*.println(**Arrays**.*toString*(a));

Arrays.asList(arraystruct).stream().forEach(s -> System.out.println(s));

Array => List

int[] i={1,2,10};

java.util.Arrays.asList(i)

(!) creates a fixed length list , doesn't allow add() or remove(), allows set()

Collection to Array

Collection.toArray()

Array => ArrayList

ArrayList<String>() al = (ArrayList) Arrays.asList(arrayofStrings);

ArrayList => Enumeration:

Collections.enumeration( structArrayList );

ArrayList => sync. list for thread-safe manipulation

List threadSafeList = Collections.synchronizedList( structArrayList );

Enumeration => collections:

Collections.enumeration()

Collections.list()

Enumeration => ArrayList:

Collections.list( request.getHeaderNames() ) .size()

int => String

String.valueOf(3); new Integer(3).toString()

Date => String

new SimpleDateFormat("w dd.MM.yy HH:mm").format(new java.util.Date())

|  |  |
| --- | --- |
| G Era designator Text AD  y Year Year 1996; 96  Y Week year Year 2009; 09  M Month in year Month July; Jul; 07  w Week in year Number 27  W Week in month Number 2  D Day in year Number 189  d Day in month Number 10  F Day of week in month Number 2  E Day name in week Text Tuesday; Tue  u Day number of week (1 = Monday, ..., 7 = Sunday) Number 1 | a Am/pm marker Text PM  H Hour in day (0-23) Number 0  k Hour in day (1-24) Number 24  K Hour in am/pm (0-11) Number 0  h Hour in am/pm (1-12) Number 12  m Minute in hour Number 30  s Second in minute Number 55  S Millisecond Number 978  z Time zone General time zone Pacific Standard Time; PST; GMT-08:00  Z Time zone RFC 822 time zone -0800  X Time zone ISO 8601 time zone -08; -0800; -08:00  To escape a string use single quote:  SimpleDateFormat("'Week'w dd.MM.yy HH:mm") |

String => Date

// 24 noiembrie 2014 => Date

Date dt=new SimpleDateFormat ("dd MMMM yyyy").parse(transactionDate.replace("noiembrie", "November"));

System.out.println(dt.toString());

String => Number

**Integer**.*parseInt*("1");

**Double**.*parseDouble*("2.345");

**Long**.*parseLong*("2");

// display current thousand and decimal separators

System.out.println(String.format("%,.2f",1000\*1000/3f));

// or using java.text.DecimalFormat

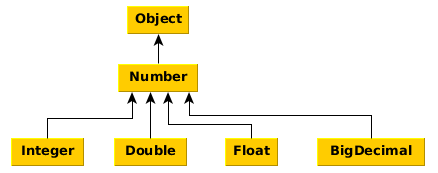
DecimalFormat().getDecimalFormatSymbols().getDecimalSeparator());

DecimalFormat().getDecimalFormatSymbols().getGroupingSeparator());

BigDecimal x = new BigDecimal(

java.text.NumberFormat.getInstance(Locale.ITALIAN)

.parse(cleanupTransactionAmount(transactionAmount)).toString());



Rounding

(**int**) 2.718% // 2

Math.round(2.718)) //3

String.format("%.2f", 2.718) //2.72

String.format("%.3s", "2.718") //2.7

Regex

String

.replaceAll ("(?i)"+ word, translation);

.split(regex)

.matches(regex)

.replaceFirst(regex)

.replace(old,new) // (!) is not using a regex object

**xml - xsd to class, xml to object**

1. generate classes from xsd (schema) using the “binding compiler” bin\xjc:

……………………………………………………………………………………………………………………………………………………………

# data-types

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Unique | Seek by Position | Insertion  Order | Thread Safe | Internal  Impl. | Default Size | Default  Resize  Rate | Retrieval  Speed/Performance |
| ArrayList  most used, alternative to Array because Array size is fixed; |  |  |  | - | Array |  | 50% | O(1) add/remove elements from the end of the container;  O(n-i)- linear- for elements from any other position;  n= number of elements;  i= index of the element; |
| Vector |  |  |  | YES | Array |  | x2 |  |
| LinkedList |  |  |  |  |  |  |  | O(1) add/remove an element at any position in constant time;  Creates an internal object for each element inserted; |
| Hashtable |  |  |  |  |  |  |  |  |
| (I) Collection  .add()  .remove()/.clear()  .iterator()  .size()  .contains() |  |  |  |  |  |  |  |  |
| (I) Set …> Collection  .get()  .add()  .iterator()  .size()  .contains() | YES |  |  |  |  |  |  |  |
| HashSet() |  |  | NO |  | HashMap |  |  |  |
| EnumSet() |  |  |  |  |  |  |  |  |
| LinkedHashSet() |  |  | YES |  |  |  |  |  |
| TreeSet() |  |  | NO  has sorting order |  |  |  |  |  |

# arrays

## concat/combine arrays/combine

void methodX (String… p){

System.arraycopy (srcArray, srcPos, destPod, srcLength);

}

**org.apache.commons.lang.ArrayUtils**

String[] arr = { "Red", "Orange", "Blue", "Brown", "Red" };

if (ArrayUtils.contains(stringArray, "Blue")){

}

int indexOfRed = ArrayUtils.indexOf(stringArray, "Red");

# org.apache.commons/utils

# org.apache.commons.lang.Validate

Validate.notNull(

Validate.notEmpty(

# org.springframework

# org.springframework.core.io.ClassPathResource;

final Resource propertiesResource = new ClassPathResource("/notifgtw/alert/" + alert + "\_" + lang + ".properties");

Validate.isTrue(propertiesResource.exists());

final Properties properties = PropertiesLoaderUtils.loadProperties(propertiesResource);

Validate.notNull(properties);

**org.springframework.core.io.Resource;**

**org.springframework.core.io.support.PropertiesLoaderUtils**

org.springframework.util.FileCopyUtils;

org.springframework.context.i18n.LocaleContextHolder;

# reflection

.**class** = class literal applicable to Class sub-types, value is Class instance of the class

Example: Object “o” has a getAttribute(String) methods:

o.getClass().getMethod("getAttribute", new Class[]{String.class}).invoke(o,"my-field");

Classes from a “default” package can’t be accessed from an explicit package (like org.apache.jasper.runtime.JspWriterImpl) , except via reflection:

String returnsString= Class.forName("uploadFile2")

.getDeclaredMethod("test")

.invoke\*(Class.forName("uploadFile2").newInstance()).toString();

***\****in this case method test() of class uploadFIle2 has no parameters, as such invoke() will have only the mandatory parameter specified; One note on the mandatory parameter “object the underlying method”, if the class is static the param is null, since there is no instance of an object to address but rather the class static “instance”;

# initialization

# 2 posibilities to create an object:

**Object** a=**new** Object();

**Object** c=**Object**.**class**.newInstance();

Default values:

|  |  |
| --- | --- |
| byte (byte) 0  short (short) 0  int 0  long 0L  float 0.0f  double 0.0d | boolean false  char \u0000  object reference null |

# one-line/compile-time initialization

# Array:

# String[] a32 = new String[]{"a","b”,"c"};

int[] a86 = new int[]{2,9,1};

# ArrayList (anonymous inner class, double brace init):

# java.util.ArrayList<String> a72

# =new java.util.ArrayList<String>() {{ add("Z"); add("X"); }};

List:

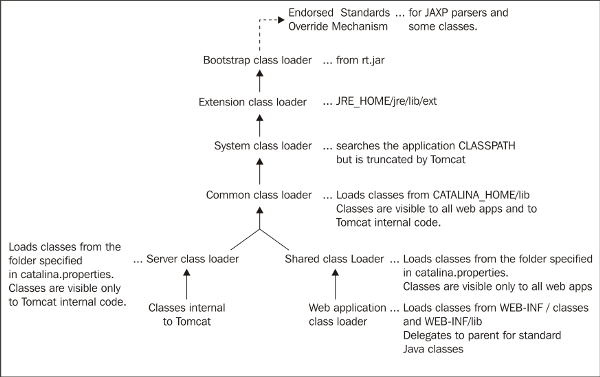
List<String> coolStringList

= Arrays.asList("A", "B", "C");

# classloader/classpath

The algorithm is delegate to parent loader, if unsuccessful child will try to locate and load the class.

Exception[[1]](#footnote-1): Java API for XML Processing Parser are loaded by the Bootstrap class loader, and so take precedence over any parser that you might have installed on your classpath;



Tomcat does not resolve classpaths in the same way as other Java programs.

## Notes

A class is considered fully qualified only when:

* package name;
* its class name;
* class loader instance;

The same class loaded by two different class loaders is treated as two distinct classes; Singletons and static classes are unique per classloader (ex Extension folder can see classes that are served up by either the Extension class loader or the Bootstrap class loader.;

Each class loader can only see the class locations that are above it in the hierarchy;

When code in a class references another class, the referenced class is loaded using the same class loader that loaded the referencing class, called its defining class loader. The defining class loader for a class can be obtained using class.getClassLoader().

## Notes Java EE

A EE compliant servlet container is required to provide a restricted environment for its web applications. Each web application is given its own class loader that preferentially loads classes in WEB-INF/classes and WEB-INF/lib; only delegate to its parent class loader when the class that is being requested is one of the standard Java classes;

System classloader = last classloader created in the JVM, uses “-classpath” switch; relying on this classloader in “container contexts” will not work (web app, ejb, frameworks …);

Current classloader = the classloader which was responsible for loading the class of the current method (current scope); to load in this conext use Class.forName() and Class.getResource() – should be used in “container context” like web app/ejb;

Context classloader = Every thread has a context class loader

Thread.currentThread().getContextClassLoader();

The class loader for the main() thread is the System class loader, which is automatically propagated down to each worker thread,

unless you intervene by invoking Thread.currentThread().setContextClassLoader()

Thread context classloaders = each thread has a context, unless created via native-code or no loader is explicitly mentioned, in which case Thread conext will inherit the Current Classloader; load in this context via Thread.setContextClassLoader()

An object can be created from a class in one thread-context then passed to a thread owned by another classloader (context);

Loading a class, with initialization flag and using a particular classloader:

Class.forName("pillarsStudy", true\*, System.class.getClassLoader())

\* If initialization is false, there is the risk that a java.lang.ExceptionInInitializerError will be thrown on 1st reference due to static code initialization;

tood

show actual search path for classes and resources:

relation between Java Naming and Directory Interface (JNDI) and classloader/context, JNDI uses context classloaders

for (java.net.URL url:

((URLClassLoader)ClassLoader.getSystemClassLoader()).getURLs()){

System.out.println(url.toString());

}

Getting classloaders :

Current thread context : Thread.currentThread().getContextClassLoader()

Any Class: ClassA.class.getClassLoader()

Library class or object: objectFromLib.getClass().getClassLoader()

System classloader: ClassLoader.getSystemClassLoader().toString()

org.springframework.util.ClassUtils;

# Servlets/Java Pages

## Java Pages Process

Jasper translation main characteristics:

* new servlet package is **org.apache.jsp;**
* almost all content is placed inside the **\_jspService (request, response**), new methods or members can be added via **<%! %>** pair (declaration area)**;**
* implicit-objects are created inside \_jspService() scope: out, page, pageContext, application, config, session,….;
* all implicit-objects are using **javax.servlet.jsp.PageContext** to further obtain references to implicit-objects;
* **<jsp:useBean** id=" " class=" "/> is also declared and initialised inside \_jspService() and fetched via PageContext instance, the bean is also added to the page context attributes, key is id tag and value is the initialization reference;

|  |  |
| --- | --- |
| basic model | specialized controller-model – idem(page) vs nonindempotent (task) controllergit |
| basic controller-model (servlet as controller, jsp as view) | command-pattern |

|  |  |
| --- | --- |
| PrintWriter = character-stream  ServletOutputStream = byte-stream | *can’t instances of both*  *PrintWriter and ServletOutputStream*  *in a single servlet method, produces java.lang.IllegalStateException* |

**Static Pillars Java**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **String** | **.valueOf()**  **.indexOf()** | **Collections** | **.list()** | **Arrays** | **.copyOf()**  **.fill()**  **.sort()** | **Types** | **Integer.class**  **Long…**  **String.class**  **Character.class**  **Date.class**  **Calendar.getInstance()**  **Math()** |

**Main objects/static pillars servlet**

getServletContext();

|  |  |
| --- | --- |
| **Move between pages/pages-and-servlets** | **Marshall data between pages/pages-and-servlets** |
| request.getRequestDispatcher(uri)  .forward(request, response);  getServletContext().getRequestDispatcher("/banner")  .include(request, response); //appends response?  sendRedirect(); // handled by the browser with 302 Redirect  pageContext.forward ("other.jsp"); // applicable from jsp | // available as ${ timestamp } in JSP  request.setAttribute("timestamp", new Date());  // available as ${bean} in JSP  request.setAttribute("bean", bean); |

**Usual**

request.getSession(true); //if Session not init, create

//<jsp:useBean id="whatever" class="com.whatever.WhateverBean"/>

//

//<%

// whatever.doSomethingWonderful();

//%>

java.net.URLEncoder.encode("http://www.sun.com")

rq.getRequestDispatcher("/WEB-INF/jsp/main.jsp").include(rq, re);

getServletContext().getRequestDispatcher("/WEB-INF/jsp/main.jsp").include(rq, re);

//execution is not-resumed

rq.getRequestDispatcher("main.jsp").forward(rq, re);

//response should not be committed (touched)

re.sendRedirect("main.jsp");

**ServletContext**

//org.apache.catalina.core.**ApplicationContext**Facade***@***581170ac

ServletContext scx=getServletContext();

// POST

String mimeType = request.getContentType();

int nBytes = request.getContentLength();

|  |  |
| --- | --- |
| **getReader()** or/ not both **getInputStream()** | |
| // text  BufferedReader buff = request.getReader(); | // raw binary  ServletInputStream sis = request.getInputStream();  int readIn = sis.readLine()  […] |

**Special tags**

<%@ directive area (import…)

<%! declaration area (define/overwrite members)

<% scriptlet tag

<% jsp:useBean

<jsp:include page="

**Special members**

**Note: all field members of a servlet/\_jsp servlet have to be concerned with thread-safety because are shared between requests (due to the usual java app-server design of thread-per-request)**

|  |  |  |  |
| --- | --- | --- | --- |
| **JSP scope** | **Servlet scope** | **desc** | **tomcat** |
| out | response.getWriter() | ref to instance of  javax.servlet.jsp.JspWriter  (<-java.io.Writer),  throws IOException | org.apache  .jasper  .runtime  .JspWriterImpl |
| session | request.getSession() | ref to instance of  javax.servlet.http.HttpSession | org.apache  .catalina  .session  .StandardSessionFacade |
| application | request.getServletContext()  getSession().getServletContext() | ref to instance of wrapper of javax.servlet.ServletContext | org.apache  .catalina  .core  .ApplicationContextFacade |
| config  1 per Servlet  only getters? | getServletConfig() | ref to instance of  javax.servlet.ServletConfig,  stores JSP engine  initialization parameters:  paths, file locations , config.  getServletName(); | org.apache  .catalina  .core  .StandardWrapperFacade |
| pageContext | N/A  relay for the other implicit objects | (<-javax.servlet.jsp. JspContext)  .getSession()  .getRequest()  .getResponse()  .getOut()  .getServletConfig()  .getServletContext()  .removeAttribute()  .forward(url) | org.apache  .jasper  .runtime  .PageContextImpl |
| page | this | this (pagename\_jsp servlet) | class generated by jasper after jsp to java class translation |
| exception |  |  |  |

ServletContext = Gloabal at web-application level, web.xml <context-param> element, programmatically add Listeners and Filters: addServlet(), addFilter(), addListener()

1 per web-app

request.getServletContext();

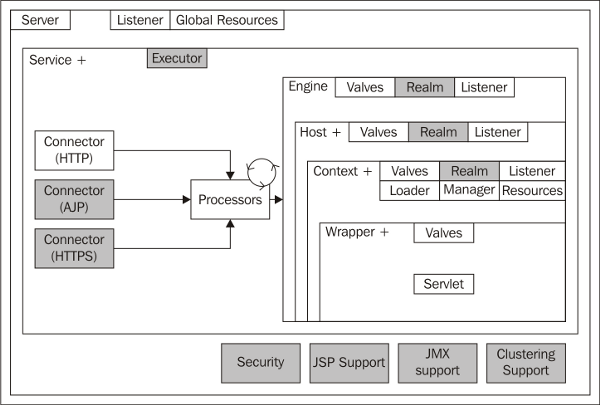
getServletConfig().getServletContext().set

ServletConfig = Specific to a certain servlet;

**How-it works**

|  |  |  |
| --- | --- | --- |
|  | GenericServlet | HttpServlet (Servlet or JSP-Converted Servlet) |
| HTTP request=> | void  service(ServletRequest req, ServletResponse res) | doMethod…  void doGet(HttpServletRequest req, HttpServletResponse resp) |
|  | | **↑↓** |
| JavaBean Business-Logic Object |

**How-it works – Tomcat**



Model “1 thread per http request” - tomcat (allowed by Java 4 New I/O APIs, non-blocking I/O)

Model “1 thread per http connection” – other application servers

Model “1 processes per request” - CGI scripts

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| HTTP request | Thread | requestthread pool | Servlet/JSP  1-instance | Native System  1-Process |
| HTTP request | Thread |
| HTTP request | Thread |
| HTTP request | Thread |

SingleThreadModel Interface

* allows for no- concurrency in the servlet’s service method;

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Session 1 | Request1 | Thread | Servlet/JSP  1 Instance | Native System  1 Process |
| Request2 | Thread |
| Session 2 | Request1 | Thread |
| Request1 | Thread |

**JSR**

JSR 315 - Servlet 3.0: Asynchronous support (ajax-and-alike request oriented)

*@WebServlet(name="myServlet", urlPatterns={"/slowprocess"},* ***asyncSupported=true****)*

*public class MyServlet extends HttpServlet {*

*public void doGet(HttpServletRequest request, HttpServletResponse response) {*

*AsyncContext aCtx = request.startAsync(request, response);*

*((Queue<AsyncContext>) request.getServletContext().getAttribute("slowWebServiceJobQueue")) ;*

asyncSupported=true => response object is not committed on method exit;

AsyncContext with application-scope, caches the request/response object pair;

*[…]*

*@WebServletContextListener*

*public class SlowWebService implements ServletContextListener {*

*public void contextInitialized(ServletContextEvent sce) {*

*Queue<AsyncContext> jobQueue = new ConcurrentLinkedQueue<AsyncContext>();*

*sce.getServletContext().setAttribute("slowWebServiceJobQueue", jobQueue);*

*// pool size matching Web services capacity*

*Executor executor = Executors.newFixedThreadPool(10);*

*while(true) {*

*if(!jobQueue.isEmpty()) {*

*final AsyncContext aCtx = jobQueue.poll();*

*executor.execute(new Runnable(){*

*public void run() {*

*ServletRequest request = aCtx.getRequest();*

*// get parameteres*

*// invoke a Web service endpoint*

*// set results*

*aCtx.forward("/result.jsp");*

*}*

*});*

*}*

*}*

*}*

*public void contextDestroyed(ServletContextEvent sce) {*

*}*

*}*

ServletContextListener handles the queue

For other events see notes-tomcat-derby.docx/Lifecycle/Events

**HTTP**

HTTP 1.0 – http connection closed after request/response;

HTTP 1.1 – introduces persistent connections, doesn’t renegotiate the TCP connection after each request;

# idiosyncratic java

**Default Package**

Pre-J2SE 1.4, default package classes could be imported in a package: *import Unfinished;*

Post-J2SE 1.4 not permited (see BUG JDK-4989710 resolution)

*Classes from the unnamed namespace are not in scope in an import declaration.*

*I believe the compiler is behaving as specified.*

*True.*

# programming practices

Use Interface type instead of Implementation type:

* strongly recommended programming practice because it gives you the flexibility to change implementations merely by changing the constructor;
* referring (to collections ) only by interface prevents you from using any nonstandard operations;

Set<String> s = new HashSet<String>();

# java/jre options

java **-XX:+PrintFlagsFinal** = what options HotSpot ended up using for running Java code;

java **-XX:+PrintFlagsInitial** = what options HotSpot used initially, before HotSpot has made its own tweaks;

Example:

java -XX:+PrintFlagsFinal -XX:+UnlockDiagnosticVMOptions -version |egrep -i -E "heapsize|permsize|version”

java **–prof** = generates a profiling dump “java.prof”;

# tools: jvisualvm, jhat…

jvisualvm

jconsole

jmc java mission control (JConsole+)

# Java2 ()

Introduction of “delegating model in a parent-child hierarchy” for classloaders

Thread context classloaders

Collections framework (designed and developed primarily by Joshua Bloch/Thinking in Java guy…)

# Java4 ()

New I/O aka NIO

Introduced Java Logging API

# Java5 (2004)

Generics

Added some Concurrent Collection classes :

ConcurrentHashMap, CopyOnWriteArrayList, BlockingQueue

**Java6 (2006)**

Annotation

New implementation of Map : NavigableMap

# Java7 (2011)

JavaXML

# Java8 (2014)

New language features aiming at conciseness:

Lambda expression

Stream API (including Collectors reduction operations like .toCollection(), .toSet()

Default methods

Improved annotation: @NonNull

DateTimeFormatter

# Java9 proposals

Process API Updates

New HTTP Client (replace java.net.HttpURLConnection, HTTP/2 support, light-weight JSON API)

# ETC

Google's open-source framework for writing, running and viewing the results of JavaMicrobenchmarks

https://code.google.com/p/caliper/

……………………………………………………………………………………………………………………………………………………………

# Annex

JSP

<http://www.mulesoft.com/tcat/tomcat-jsp>

<https://www.packtpub.com/books/content/overview-tomcat-6-servlet-container-part-1>

http://www.examulator.com/moodle/mod/resource/index.php?id=5

OC4J Summary of What to Know About Servlets

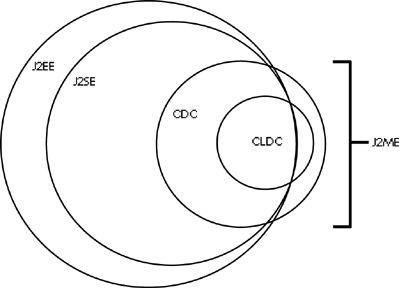
<http://otndnld.oracle.co.jp/document/products/as10g/101300/B25221_03/web.1013/b14426/overview.htm#BABGJGEE>

cheet-sheet

<http://introcs.cs.princeton.edu/java/11cheatsheet/>

quick-reference collection

<http://www.javaprogrammingforums.com/java-programming-tutorials/15449-64-java-cheat-sheets-ref-cards.html>



1. The Endorsed Standards Override Mechanism lets you place overrides to certain classes (CORBA and JAXP classes) in the JAVA\_HOME/lib/endorsed folder. [↑](#footnote-ref-1)