	Line of Code	Code	Issue	Comments	
	90	private static final ResourceBundle rb = log.getResourceBundle();	1	rb does not mean anything	
	151	private LifecycleSupport lifecycle = new LifecycleSupport(this);	1	"lifecycle" does not suggest that is a lifecycleSupport	
	168	protected int debug = 0;	1	"debug" does not suggest that is a debug level variable	
	200	private NotificationBroadcasterSupport broadcaster = null;	1	"broadcaster" does not suggest that is a NOTIFICATION broadcaster.	
	279	public int getDebug()	1	does not suggest that is the debug level	
	291	public void setDebug(int debug)	1	does not suggest that is the debug level	

	418,
	419,
	420,
	421,
	422,
	423,
	424,
	425,
	426,
	427
		public ObjectName[] getConnectorNames() {
		ObjectName results[] = new ObjectName[connectors.length];
		for(int i=0; i<results.length; i++) {
		// if it's a coyote connector
		//if(connectors[i] instanceof CoyoteConnector) {
		// results[i]=((CoyoteConnector)connectors[i]).getJmxName();
		//}
		}
		return results;
		}
			Any other problem...	method that does nothing. it returns only a empty array!	

	745	protected ObjectName oname;	1	oname is meainingless	
	89	private static final Logger log = StandardServer.log;	7	final but not uppercase separated by '_'	
	90	private static final ResourceBundle rb = log.getResourceBundle();	7	final but not uppercase separated by '_'	
	138, 139	private static final String info = "org.apache.catalina.core.StandardService/1.0";	7	final but not uppercase separated by '_'	
	182	protected final Object connectorsMonitor = new Object();	7	final but not uppercase separated by '_'	

	488
	489
		if (j < 0)
		return;
			11		

	509
	510
		if (i != j)
		results[k++] = connectors[i];
			11		

	596
	597
		if(! initialized)
		init();
			11		

	620
	621
		if (connectors[i] instanceof Lifecycle)
		((Lifecycle) connectors[i]).start();
			11		

	660
	661
		if (connectors[i] instanceof Lifecycle)
		((Lifecycle) connectors[i]).stop();
			11		

	703	String msg = MessageFormat.format(rb.getString(ERROR_REGISTER_SERVICE_EXCEPTION), domain);	13	it's pratical in this line of code to have a length less than 80 chars	
	739	String msg = MessageFormat.format(rb.getString(FAILED_SERVICE_INIT_EXCEPTION), domain);	13	it's pratical in this line of code to have a length less than 80 chars	
	85 86	public class StandardService implements Lifecycle, Service	15	line break occurs after a space	
	88-130	all lines	17	all these lines have an offset of one space character from their level of indentation.	
	475	//public void removeConnector(Connector connector) {	19		

	492
	493
	494
	495
	496
	497
	498
		/*if (started && (connectors[j] instanceof Lifecycle)) {
		try {
		((Lifecycle) connectors[j]).stop();
		} catch (LifecycleException e) {
		log.error("Connector.stop", e);
		}
		}*/
			19	SJSAS 6231069	

	503 504	/*connectors[j].setContainer(null); connector.setService(null);*/	19	SJSAS 6231069	
	The class file only contains one public class. so it's all ok for issue 21				
	96	public static final String SERVICE_STARTED = "AS-WEB-CORE-00251";	23	no javadoc for public static attribute	
	102	public static final String STARTING_SERVICE = "AS-WEB-CORE-00252";	23	no javadoc for public static attribute	
	108	public static final String STOPPING_SERVICE = "AS-WEB-CORE-00253";	23	no javadoc for public static attribute	
	114	public static final String SERVICE_HAS_BEEN_INIT = "AS-WEB-CORE-00254";	23	no javadoc for public static attribute	
	122	public static final String ERROR_REGISTER_SERVICE_EXCEPTION = "AS-WEB-CORE-00255";	23	no javadoc for public static attribute	
	130	public static final String FAILED_SERVICE_INIT_EXCEPTION = "AS-WEB-CORE-00256";	23	no javadoc for public static attribute	
	268	public ObjectName getContainerName()	23	no javadoc for public method	
	279	public int getDebug()	23	better to use "@return" javadoc command instead of writing in the description field	
	305	public String getInfo()	23	better to use "@return" javadoc command instead of writing in the description field	
	315	public String getName()	23	better to use "@return" javadoc command instead of writing in the description field	
	337	public Server getServer()	23	better to use "@return" javadoc command instead of writing in the description field	
	358	public NotificationBroadcasterSupport getBroadcaster()	23	better to use "@return" javadoc command instead of writing in the description field	
	418	public ObjectName[] getConnectorNames()	23	no javadoc for public method	
	445	public Connector[] findConnectors()	23	better to use "@return" javadoc command instead of writing in the description field	
	455	public Connector findConnector(String name)	23	better to use "@return" javadoc command instead of writing in the description field, no "@param" javadoc field for tne "String name" parameter	
	476	public void removeConnector(Connector connector) throws LifecycleException	23	no "@throws" javadoc field for the exception "LifecycleException"	
	536	public String toString()	23	better to use "@return" javadoc command instead of writing in the description field	
	563	public List<LifecycleListener> findLifecycleListeners()	23	better to use "@return" javadoc command instead of writing in the description field	
	684 685	public void initialize() throws LifecycleException	23	no "@throws" javadoc field for the exception "LifecycleException"	
	725	public void destroy() throws LifecycleException	23	no javadoc for public method	
	735	public void init()	23	no javadoc for public method	
	747	public ObjectName getObjectName()	23	no javadoc for public method	
	751	public String getDomain()	23	no javadoc for public method	
	89	private static final Logger log = StandardServer.log;	25	private static attribute stated before a public static one	
	90	private static final ResourceBundle rb = log.getResourceBundle();	25	private static attribute stated before a public static one	
	145	private String name = null;	25	private attribute stated before a protected one	
	151	private LifecycleSupport lifecycle = new LifecycleSupport(this);	25	private attribute stated before a protected one	
	157	private Server server = null;	25	private attribute stated before a protected one	
	162	private boolean started = false;	25	private attribute stated before a protected one	
	744	protected String domain;	25	protected attribute not in the attributes section	
	745	protected ObjectName oname;	25	protected attribute not in the attributes section	
add	391	Connector results[] = new Connector[connectors.length + 1];	33	not declared at the top of the block	
	506	int k = 0;	33	not declared at the top of the block	
	507	Connector results[] = new Connector[connectors.length - 1];	33	not declared at the top of the block	
		Note on 39: not everytime a new array is created his elements are initialized using the constructor, but in these cases it's not a problem because the elements are copied from another array that was initialized before.			
		Note on 51: an implicit conversion is intended as a conversion where an object or a variable is implicitly converted by the compiler			
add	390	connector.setService(this);	51	"this" is of the type StandardService, where the parameter of setService is "Service"	riportare javadoc
add	392	System.arraycopy(connectors, 0, results, 0, connectors.length);	51	connectors and result are of the type "Connector" where the types of the first and the third parameters of System.arraycopy are "Object"	riportare javadoc
add	400	log.log(Level.SEVERE, "Connector.initialize", e);	51	the type of "e" is "LifecycleException" where the type of the third parameter of Logger.log() is "Throwable"	riportare javadoc
add	408	log.log(Level.SEVERE, "Connector.start", e);	51	the type of "e" is "LifecycleException" where the type of the third parameter of Logger.log() is "Throwable"	riportare javadoc
add	413	support.firePropertyChange("connector", null, connector);	51	type of the the third parameter is "Object" and "connector" is of the type "Connector"	riportare javadoc
	515	support.firePropertyChange("connector", connector, null);	51	type of the second parameter is "Object" and "connector" is of the type "Connector"	riportare javadoc
	603	log.log(Level.INFO, STARTING_SERVICE, this.name);	51	type of the third parameter is "Object" where this.name is String.	riportare javadoc
	653	log.log(Level.INFO, STOPPING_SERVICE, this.name);	51	type of the third parameter is "Object" where this.name is String.	riportare javadoc
	703	String msg = MessageFormat.format(rb.getString(ERROR_REGISTER_SERVICE_EXCEPTION), domain);	51	"domain" is a String and it's implicitly converted into "Object"	riportare javadoc
	704	log.log(Level.SEVERE, msg, e);	51	"e" is an Exception and it's implicitly converted into "Object"	riportare javadoc
	713	ServerFactory.getServer().addService(this);	51	"this" is a StandardService and it's implocitly converted into "Service"	riportare javadoc
	702	catch (Exception e)	53	"MalformedObjectNameException" should be caught instead of "Exception"	