

# EJB Examples

Example `ejb-examples` can be browsed at <https://github.com/apache/tomee/tree/master/examples/ejb-examples>

**Help us document this example! Click the blue pencil icon in the upper right to edit this page.**

## AnnotatedEJB

```
package org.superbiz.servlet;

import javax.annotation.Resource;
import javax.ejb.LocalBean;
import javax.ejb.Stateless;
import javax.sql.DataSource;

@Stateless
@LocalBean
public class AnnotatedEJB implements AnnotatedEJBLocal, AnnotatedEJBRemote {
    @Resource
    private DataSource ds;

    private String name = "foo";

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public DataSource getDs() {
        return ds;
    }

    public void setDs(DataSource ds) {
        this.ds = ds;
    }

    public String toString() {
        return "AnnotatedEJB[name=" + name + "]";
    }
}
```

# AnnotatedEJBLocal

```
package org.superbiz.servlet;

import javax.ejb.Local;
import javax.sql.DataSource;

@Local
public interface AnnotatedEJBLocal {
    String getName();

    void setName(String name);

    DataSource getDs();

    void setDs(DataSource ds);
}
```

# AnnotatedEJBRemote

```
package org.superbiz.servlet;

import javax.ejb.Remote;

@Remote
public interface AnnotatedEJBRemote {
    String getName();

    void setName(String name);
}
```

# AnnotatedServlet

```
package org.superbiz.servlet;

import javax.annotation.Resource;
import javax.ejb.EJB;
import javax.naming.InitialContext;
import javax.naming.NamingException;
import javax.servlet.ServletException;
import javax.servlet.ServletOutputStream;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.sql.DataSource;
```

```

import java.io.IOException;

public class AnnotatedServlet extends HttpServlet {
    @EJB
    private AnnotatedEJBLocal localEJB;

    @EJB
    private AnnotatedEJBRemote remoteEJB;

    @EJB
    private AnnotatedEJB localbeanEJB;

    @Resource
    private DataSource ds;

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        response.setContentType("text/plain");
        ServletOutputStream out = response.getOutputStream();

        out.println("LocalBean EJB");
        out.println("@EJB=" + localbeanEJB);
        if (localbeanEJB != null) {
            out.println("@EJB.getName()" + localbeanEJB.getName());
            out.println("@EJB.getDs()" + localbeanEJB.getDs());
        }
        out.println("JNDI=" + lookupField("localbeanEJB"));
        out.println();

        out.println("Local EJB");
        out.println("@EJB=" + localEJB);
        if (localEJB != null) {
            out.println("@EJB.getName()" + localEJB.getName());
            out.println("@EJB.getDs()" + localEJB.getDs());
        }
        out.println("JNDI=" + lookupField("localEJB"));
        out.println();

        out.println("Remote EJB");
        out.println("@EJB=" + remoteEJB);
        if (remoteEJB != null) {
            out.println("@EJB.getName()" + remoteEJB.getName());
        }
        out.println("JNDI=" + lookupField("remoteEJB"));
        out.println();

        out.println("DataSource");
        out.println("@Resource=" + ds);
        out.println("JNDI=" + lookupField("ds"));
    }
}

```

```

    }

    private Object lookupField(String name) {
        try {
            return new InitialContext().lookup("java:comp/env/" + getClass().getName()
+ "/" + name);
        } catch (NamingException e) {
            return null;
        }
    }
}

```

## ClientHandler

```

package org.superbiz.servlet;

import javax.xml.ws.handler.Handler;
import javax.xml.ws.handler.MessageContext;

public class ClientHandler implements Handler {
    public boolean handleMessage(MessageContext messageContext) {
        WebserviceServlet.write("    ClientHandler handleMessage");
        return true;
    }

    public void close(MessageContext messageContext) {
        WebserviceServlet.write("    ClientHandler close");
    }

    public boolean handleFault(MessageContext messageContext) {
        WebserviceServlet.write("    ClientHandler handleFault");
        return true;
    }
}

```

## HelloEjb

```

package org.superbiz.servlet;

import javax.jws.WebService;

@WebService(targetNamespace = "http://examples.org/wsd1")
public interface HelloEjb {
    String hello(String name);
}

```

# HelloEjbService

```
package org.superbiz.servlet;

import javax.ejb.Stateless;
import javax.jws.HandlerChain;
import javax.jws.WebService;

@WebService(
    portName = "HelloEjbPort",
    serviceName = "HelloEjbService",
    targetNamespace = "http://examples.org/wsdl",
    endpointInterface = "org.superbiz.servlet.HelloEjb"
)
@HandlerChain(file = "server-handlers.xml")
@Stateless
public class HelloEjbService implements HelloEjb {
    public String hello(String name) {
        WebserviceServlet.write("HelloEjbService hello(" + name + ")")
    };
    if (name == null) name = "World";
    return "Hello " + name + " from EJB Webservice!";
}
}
```

# HelloPojo

```
package org.superbiz.servlet;

import javax.jws.WebService;

@WebService(targetNamespace = "http://examples.org/wsdl")
public interface HelloPojo {
    String hello(String name);
}
```

# HelloPojoService

```

package org.superbiz.servlet;

import javax.ws.HandlerChain;
import javax.ws.WebService;

@WebService(
    portName = "HelloPojoPort",
    serviceName = "HelloPojoService",
    targetNamespace = "http://examples.org/wsdl",
    endpointInterface = "org.superbiz.servlet.HelloPojo"
)
@HandlerChain(file = "server-handlers.xml")
public class HelloPojoService implements HelloPojo {
    public String hello(String name) {
        WebserviceServlet.write("                HelloPojoService hello(" + name + ")
");
        if (name == null) name = "World";
        return "Hello " + name + " from Pojo Webservice!";
    }
}

```

## JndiServlet

```

package org.superbiz.servlet;

import javax.naming.Context;
import javax.naming.InitialContext;
import javax.naming.NameClassPair;
import javax.naming.NamingException;
import javax.servlet.ServletException;
import javax.servlet.ServletOutputStream;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.util.Collections;
import java.util.Map;
import java.util.TreeMap;

public class JndiServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        response.setContentType("text/plain");
        ServletOutputStream out = response.getOutputStream();

        Map<String, Object> bindings = new TreeMap<String, Object>(String
.CASE_INSENSITIVE_ORDER);
        try {

```

```

        Context context = (Context) new InitialContext().lookup("java:comp/");
        addBindings("", bindings, context);
    } catch (NamingException e) {
        throw new ServletException(e);
    }

    out.println("JNDI Context:");
    for (Map.Entry<String, Object> entry : bindings.entrySet()) {
        if (entry.getValue() != null) {
            out.println(" " + entry.getKey() + "=" + entry.getValue());
        } else {
            out.println(" " + entry.getKey());
        }
    }
}

private void addBindings(String path, Map<String, Object> bindings, Context
context) {
    try {
        for (NameClassPair pair : Collections.list(context.list(""))) {
            String name = pair.getName();
            String className = pair.getClassName();
            if ("org.apache.naming.resources.FileDirContext$FileResource".equals
(className)) {
                bindings.put(path + name, "<file>");
            } else {
                try {
                    Object value = context.lookup(name);
                    if (value instanceof Context) {
                        Context nextedContext = (Context) value;
                        bindings.put(path + name, "");
                        addBindings(path + name + "/", bindings, nextedContext);
                    } else {
                        bindings.put(path + name, value);
                    }
                } catch (NamingException e) {
                    // lookup failed
                    bindings.put(path + name, "ERROR: " + e.getMessage());
                }
            }
        }
    } catch (NamingException e) {
        bindings.put(path, "ERROR: list bindings threw an exception: " + e
.getMessage());
    }
}
}

```



# JpaBean

```
package org.superbiz.servlet;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;

@Entity
public class JpaBean {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "id")
    private int id;

    @Column(name = "name")
    private String name;

    public int getId() {
        return id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String toString() {
        return "[JpaBean id=" + id + ", name=" + name + "]";
    }
}
```

# JpaServlet

```
package org.superbiz.servlet;

import javax.persistence.EntityManager;
import javax.persistence.EntityManagerFactory;
import javax.persistence.EntityTransaction;
import javax.persistence.PersistenceUnit;
import javax.persistence.Query;
```

```

import javax.servlet.ServletException;
import javax.servlet.ServletOutputStream;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;

public class JpaServlet extends HttpServlet {
    @PersistenceUnit(name = "jpa-example")
    private EntityManagerFactory emf;

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        response.setContentType("text/plain");
        ServletOutputStream out = response.getOutputStream();

        out.println("@PersistenceUnit=" + emf);

        EntityManager em = emf.createEntityManager();
        EntityTransaction transaction = em.getTransaction();
        transaction.begin();

        JpaBean jpaBean = new JpaBean();
        jpaBean.setName("JpaBean");
        em.persist(jpaBean);

        transaction.commit();
        transaction.begin();

        Query query = em.createQuery("SELECT j FROM JpaBean j WHERE j.name='JpaBean'");
        jpaBean = (JpaBean) query.getSingleResult();
        out.println("Loaded " + jpaBean);

        em.remove(jpaBean);

        transaction.commit();
        transaction.begin();

        query = em.createQuery("SELECT count(j) FROM JpaBean j WHERE j.name='JpaBean'");
        int count = ((Number) query.getSingleResult()).intValue();
        if (count == 0) {
            out.println("Removed " + jpaBean);
        } else {
            out.println("ERROR: unable to remove" + jpaBean);
        }

        transaction.commit();
    }
}

```

```
}
```

## ResourceBean

```
package org.superbiz.servlet;

public class ResourceBean {
    private String value;

    public String getValue() {
        return value;
    }

    public void setValue(String value) {
        this.value = value;
    }

    public String toString() {
        return "[ResourceBean " + value + "]";
    }
}
```

## RunAsServlet

```
package org.superbiz.servlet;

import javax.ejb.EJB;
import javax.ejb.EJBAccessException;
import javax.servlet.ServletException;
import javax.servlet.ServletOutputStream;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.security.Principal;

public class RunAsServlet extends HttpServlet {
    @EJB
    private SecureEJBLocal secureEJBLocal;

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        response.setContentType("text/plain");
        ServletOutputStream out = response.getOutputStream();

        out.println("Servlet");
    }
}
```

```

Principal principal = request.getUserPrincipal();
if (principal != null) {
    out.println("Servlet.getUserPrincipal()=" + principal + " [" + principal
.getName() + "]");
} else {
    out.println("Servlet.getUserPrincipal()=<null>");
}
out.println("Servlet.isCallerInRole(\"user\")=" + request.isUserInRole("user"
));
out.println("Servlet.isCallerInRole(\"manager\")=" + request.isUserInRole(
"manager"));
out.println("Servlet.isCallerInRole(\"fake\")=" + request.isUserInRole("fake"
));
out.println();

out.println("@EJB=" + secureEJBLocal);
if (secureEJBLocal != null) {
    principal = secureEJBLocal.getCallerPrincipal();
    if (principal != null) {
        out.println("@EJB.getCallerPrincipal()=" + principal + " [" +
principal.getName() + "]");
    } else {
        out.println("@EJB.getCallerPrincipal()=<null>");
    }
    out.println("@EJB.isCallerInRole(\"user\")=" + secureEJBLocal
.isCallerInRole("user"));
    out.println("@EJB.isCallerInRole(\"manager\")=" + secureEJBLocal
.isCallerInRole("manager"));
    out.println("@EJB.isCallerInRole(\"fake\")=" + secureEJBLocal
.isCallerInRole("fake"));

    try {
        secureEJBLocal.allowUserMethod();
        out.println("@EJB.allowUserMethod() ALLOWED");
    } catch (EJBAccessException e) {
        out.println("@EJB.allowUserMethod() DENIED");
    }

    try {
        secureEJBLocal.allowManagerMethod();
        out.println("@EJB.allowManagerMethod() ALLOWED");
    } catch (EJBAccessException e) {
        out.println("@EJB.allowManagerMethod() DENIED");
    }

    try {
        secureEJBLocal.allowFakeMethod();
        out.println("@EJB.allowFakeMethod() ALLOWED");
    } catch (EJBAccessException e) {
        out.println("@EJB.allowFakeMethod() DENIED");
    }
}

```

```
        try {
            secureEJBLocal.denyAllMethod();
            out.println("@EJB.denyAllMethod() ALLOWED");
        } catch (EJBAccessException e) {
            out.println("@EJB.denyAllMethod() DENIED");
        }
    }
    out.println();
}
}
```

## SecureEJB

```

package org.superbiz.servlet;

import javax.annotation.Resource;
import javax.annotation.security.DeclareRoles;
import javax.annotation.security.DenyAll;
import javax.annotation.security.RolesAllowed;
import javax.ejb.SessionContext;
import javax.ejb.Stateless;
import java.security.Principal;

@Stateless
@DeclareRoles({"user", "manager", "fake"})
public class SecureEJB implements SecureEJBLocal {
    @Resource
    private SessionContext context;

    public Principal getCallerPrincipal() {
        return context.getCallerPrincipal();
    }

    public boolean isCallerInRole(String role) {
        return context.isCallerInRole(role);
    }

    @RolesAllowed("user")
    public void allowUserMethod() {
    }

    @RolesAllowed("manager")
    public void allowManagerMethod() {
    }

    @RolesAllowed("fake")
    public void allowFakeMethod() {
    }

    @DenyAll
    public void denyAllMethod() {
    }

    public String toString() {
        return "SecureEJB[user=" + getCallerPrincipal() + "]";
    }
}

```

## SecureEJBLocal

```

package org.superbiz.servlet;

import javax.ejb.Local;
import java.security.Principal;

@Local
public interface SecureEJBLocal {
    Principal getCallerPrincipal();

    boolean isCallerInRole(String role);

    void allowUserMethod();

    void allowManagerMethod();

    void allowFakeMethod();

    void denyAllMethod();
}

```

## SecureServlet

```

package org.superbiz.servlet;

import javax.ejb.EJB;
import javax.ejb.EJBAccessException;
import javax.servlet.ServletException;
import javax.servlet.ServletOutputStream;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.security.Principal;

public class SecureServlet extends HttpServlet {
    @EJB
    private SecureEJBLocal secureEJBLocal;

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException {
        response.setContentType("text/plain");
        ServletOutputStream out = response.getOutputStream();

        out.println("Servlet");
        Principal principal = request.getUserPrincipal();
        if (principal != null) {
            out.println("Servlet.getUserPrincipal()=" + principal + " [" + principal
.getName() + "]");
        }
    }
}

```

```

    } else {
        out.println("Servlet.getUserPrincipal()=<null>");
    }
    out.println("Servlet.isCallerInRole(\"user\")=" + request.isUserInRole("user"
));
    out.println("Servlet.isCallerInRole(\"manager\")=" + request.isUserInRole(
"manager"));
    out.println("Servlet.isCallerInRole(\"fake\")=" + request.isUserInRole("fake"
));
    out.println();

    out.println("@EJB=" + secureEJBLocal);
    if (secureEJBLocal != null) {
        principal = secureEJBLocal.getCallerPrincipal();
        if (principal != null) {
            out.println("@EJB.getCallerPrincipal()=" + principal + " [" +
principal.getName() + "]");
        } else {
            out.println("@EJB.getCallerPrincipal()=<null>");
        }
        out.println("@EJB.isCallerInRole(\"user\")=" + secureEJBLocal
.isCallerInRole("user"));
        out.println("@EJB.isCallerInRole(\"manager\")=" + secureEJBLocal
.isCallerInRole("manager"));
        out.println("@EJB.isCallerInRole(\"fake\")=" + secureEJBLocal
.isCallerInRole("fake"));

        try {
            secureEJBLocal.allowUserMethod();
            out.println("@EJB.allowUserMethod() ALLOWED");
        } catch (EJBAccessException e) {
            out.println("@EJB.allowUserMethod() DENIED");
        }

        try {
            secureEJBLocal.allowManagerMethod();
            out.println("@EJB.allowManagerMethod() ALLOWED");
        } catch (EJBAccessException e) {
            out.println("@EJB.allowManagerMethod() DENIED");
        }

        try {
            secureEJBLocal.allowFakeMethod();
            out.println("@EJB.allowFakeMethod() ALLOWED");
        } catch (EJBAccessException e) {
            out.println("@EJB.allowFakeMethod() DENIED");
        }

        try {
            secureEJBLocal.denyAllMethod();
            out.println("@EJB.denyAllMethod() ALLOWED");
        }
    }
}

```



```

        } catch (EJBAccessException e) {
            out.println("@EJB.denyAllMethod() DENIED");
        }
    }
    out.println();
}
}

```

## ServerHandler

```

package org.superbiz.servlet;

import javax.xml.ws.handler.Handler;
import javax.xml.ws.handler.MessageContext;

public class ServerHandler implements Handler {
    public boolean handleMessage(MessageContext messageContext) {
        WebServiceServlet.write("        ServerHandler handleMessage");
        return true;
    }

    public void close(MessageContext messageContext) {
        WebServiceServlet.write("        ServerHandler close");
    }

    public boolean handleFault(MessageContext messageContext) {
        WebServiceServlet.write("        ServerHandler handleFault");
        return true;
    }
}

```

## WebServiceImpl

```

package org.superbiz.servlet;

import javax.xml.ws.Service;
import java.io.PrintStream;
import java.net.URL;

public class WebServiceImpl {
    /**
     * Unfortunately, to run this example with CXF you need to have a HUGE class path.
     This
     * is just what is required to run CXF:
     * <p>
     * jaxb-api-2.0.jar
     */
}

```

```

* jaxb-impl-2.0.3.jar
* <p/>
* saaj-api-1.3.jar
* saaj-impl-1.3.jar
* <p/>
* <p/>
* cxf-api-2.0.2-incubator.jar
* cxf-common-utilities-2.0.2-incubator.jar
* cxf-rt-bindings-soap-2.0.2-incubator.jar
* cxf-rt-core-2.0.2-incubator.jar
* cxf-rt-databinding-jaxb-2.0.2-incubator.jar
* cxf-rt-frontend-jaxws-2.0.2-incubator.jar
* cxf-rt-frontend-simple-2.0.2-incubator.jar
* cxf-rt-transport-http-jetty-2.0.2-incubator.jar
* cxf-rt-transport-http-2.0.2-incubator.jar
* cxf-tools-common-2.0.2-incubator.jar
* <p/>
* geronimo-activation_1.1_spec-1.0.jar
* geronimo-annotation_1.0_spec-1.1.jar
* geronimo-ejb_3.0_spec-1.0.jar
* geronimo-jpa_3.0_spec-1.1.jar
* geronimo-servlet_2.5_spec-1.1.jar
* geronimo-stax-api_1.0_spec-1.0.jar
* jaxws-api-2.0.jar
* axis2-jws-api-1.3.jar
* <p/>
* wsdl4j-1.6.1.jar
* xml-resolver-1.2.jar
* XmlSchema-1.3.1.jar
*/
public static void main(String[] args) throws Exception {
    PrintStream out = System.out;

    Service helloPojoService = Service.create(new URL("http://localhost:8080/ejb-
examples/hello?wsdl"), null);
    HelloPojo helloPojo = helloPojoService.getPort(HelloPojo.class);
    out.println();
    out.println("Pojo Webservice");
    out.println("    helloPojo.hello(\"Bob\")=" + helloPojo.hello("Bob"));
    out.println("    helloPojo.hello(null)=" + helloPojo.hello(null));
    out.println();

    Service helloEjbService = Service.create(new URL(
"http://localhost:8080/HelloEjbService?wsdl"), null);
    HelloEjb helloEjb = helloEjbService.getPort(HelloEjb.class);
    out.println();
    out.println("EJB Webservice");
    out.println("    helloEjb.hello(\"Bob\")=" + helloEjb.hello("Bob"));
    out.println("    helloEjb.hello(null)=" + helloEjb.hello(null));
    out.println();
}

```

```
}
```

# WebServiceImplServlet

```
package org.superbiz.servlet;

import javax.ws.HandlerChain;
import javax.servlet.ServletException;
import javax.servlet.ServletOutputStream;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.xml.ws.WebServiceRef;
import java.io.IOException;

public class WebServiceImplServlet extends HttpServlet {

    @WebServiceRef
    @HandlerChain(file = "client-handlers.xml")
    private HelloPojo helloPojo;

    @WebServiceRef
    @HandlerChain(file = "client-handlers.xml")
    private HelloEjb helloEjb;

    protected void doGet(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
        response.setContentType("text/plain");
        ServletOutputStream out = response.getOutputStream();

        OUT = out;
        try {
            out.println("Pojo Webservice");
            out.println("    helloPojo.hello(\"Bob\")=" + helloPojo.hello("Bob"));
            out.println();
            out.println("    helloPojo.hello(null)=" + helloPojo.hello(null));
            out.println();
            out.println("EJB Webservice");
            out.println("    helloEjb.hello(\"Bob\")=" + helloEjb.hello("Bob"));
            out.println();
            out.println("    helloEjb.hello(null)=" + helloEjb.hello(null));
            out.println();
        } finally {
            OUT = out;
        }
    }

    private static ServletOutputStream OUT;
```

```

public static void write(String message) {
    try {
        ServletOutputStream out = OUT;
        out.println(message);
    } catch (Exception e) {
        e.printStackTrace();
    }
}
}

```

## persistence.xml

```

<persistence xmlns="http://java.sun.com/xml/ns/persistence" version="1.0">
  <persistence-unit transaction-type="RESOURCE_LOCAL" name="jpa-example">
    <jta-data-source>java:openejb/Connector/Default JDBC Database</jta-data-source>
    <non-jta-data-source>java:openejb/Connector/Default Unmanaged JDBC Database</non-
jta-data-source>
    <class>org.superbiz.servlet.JpaBean</class>

```

```

    <properties>
      <property name="openjpa.jdbc.SynchronizeMappings"
value="buildSchema(ForeignKeys=true)"/>
    </properties>
  </persistence-unit>
</persistence>

```

## client-handlers.xml

```

<jws:handler-chains xmlns:jws="http://java.sun.com/xml/ns/javaee">
  <jws:handler-chain>
    <jws:handler>
      <jws:handler-name>ClientHandler</jws:handler-name>
      <jws:handler-class>org.superbiz.servlet.ClientHandler</jws:handler-class>
    </jws:handler>
  </jws:handler-chain>
</jws:handler-chains>

```

## server-handlers.xml

```

<jws:handler-chains xmlns:jws="http://java.sun.com/xml/ns/javaee">
  <jws:handler-chain>
    <jws:handler>
      <jws:handler-name>ServerHandler</jws:handler-name>
      <jws:handler-class>org.superbiz.servlet.ServerHandler</jws:handler-class>
    </jws:handler>
  </jws:handler-chain>
</jws:handler-chains>

```

## context.xml

```

<Context>
  <!-- This only works if the context is installed under the correct name -->
  <Realm className="org.apache.catalina.realm.MemoryRealm"
    pathname="webapps/ejb-examples-1.0-SNAPSHOT/WEB-INF/tomcat-users.xml"/>

```

```

<Environment
  name="context.xml/environment"
  value="ContextString"
  type="java.lang.String"/>
<Resource
  name="context.xml/resource"
  auth="Container"
  type="org.superbiz.servlet.ResourceBean"
  factory="org.apache.naming.factory.BeanFactory"
  value="ContextResource"/>
<ResourceLink
  name="context.xml/resource-link"
  global="server.xml/environment"
  type="java.lang.String"/>

```

```

<!-- web.xml resources -->
<Resource
    name="web.xml/resource-env-ref"
    auth="Container"
    type="org.superbiz.servlet.ResourceBean"
    factory="org.apache.naming.factory.BeanFactory"
    value="ContextResourceEnvRef"/>
<Resource
    name="web.xml/resource-ref"
    auth="Container"
    type="org.superbiz.servlet.ResourceBean"
    factory="org.apache.naming.factory.BeanFactory"
    value="ContextResourceRef"/>
<ResourceLink
    name="web.xml/resource-link"
    global="server.xml/environment"
    type="java.lang.String"/>
</Context>

```

## jetty-web.xml

```

<Configure class="org.eclipse.jetty.webapp.WebAppContext">
    <Get name="securityHandler">
        <Set name="loginService">
            <New class="org.eclipse.jetty.security.HashLoginService">
                <Set name="name">Test Realm</Set>
                <Set name="config"><SystemProperty name="jetty.home"
default="."/>/etc/realm.properties
                </Set>
            </New>
        </Set>
    </Get>
</Configure>

```

## tomcat-users.xml

```

<tomcat-users>
    <user name="manager" password="manager" roles="manager,user"/>
    <user name="user" password="user" roles="user"/>
</tomcat-users>

```

# web.xml

```
<web-app xmlns="http://java.sun.com/xml/ns/javaee"
         xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd"
         metadata-complete="false"
         version="2.5">
```

```
<display-name>OpenEJB Servlet Examples</display-name>
```

```
<servlet>
  <servlet-name>AnnotatedServlet</servlet-name>
  <servlet-class>org.superbiz.servlet.AnnotatedServlet</servlet-class>
</servlet>
```

```
<servlet-mapping>
  <servlet-name>AnnotatedServlet</servlet-name>
  <url-pattern>/annotated/*</url-pattern>
</servlet-mapping>
```

```
<servlet>
  <servlet-name>JpaServlet</servlet-name>
  <servlet-class>org.superbiz.servlet.JpaServlet</servlet-class>
</servlet>
```

```
<servlet-mapping>
  <servlet-name>JpaServlet</servlet-name>
  <url-pattern>/jpa/*</url-pattern>
</servlet-mapping>
```

```
<servlet>
  <servlet-name>JndiServlet</servlet-name>
  <servlet-class>org.superbiz.servlet.JndiServlet</servlet-class>
</servlet>
```

```
<servlet-mapping>
  <servlet-name>JndiServlet</servlet-name>
  <url-pattern>/jndi/*</url-pattern>
</servlet-mapping>
```

```
<servlet>
  <servlet-name>RunAsServlet</servlet-name>
  <servlet-class>org.superbiz.servlet.RunAsServlet</servlet-class>
  <run-as>
    <role-name>fake</role-name>
  </run-as>
</servlet>
```

```
<servlet-mapping>
  <servlet-name>RunAsServlet</servlet-name>
  <url-pattern>/runas/*</url-pattern>
</servlet-mapping>
```

```
<servlet>
  <servlet-name>SecureServlet</servlet-name>
  <servlet-class>org.superbiz.servlet.SecureServlet</servlet-class>
</servlet>
```

```
<servlet-mapping>
  <servlet-name>SecureServlet</servlet-name>
  <url-pattern>/secure/*</url-pattern>
</servlet-mapping>
```

```
<security-constraint>
  <web-resource-collection>
    <web-resource-name>Secure Area</web-resource-name>
    <url-pattern>/secure/*</url-pattern>
    <url-pattern>/runas/*</url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <role-name>user</role-name>
  </auth-constraint>
</security-constraint>
```



```
<servlet>
  <servlet-name>WebserviceServlet</servlet-name>
  <servlet-class>org.superbiz.servlet.WebserviceServlet</servlet-class>
</servlet>
```

```
<servlet-mapping>
  <servlet-name>WebserviceServlet</servlet-name>
  <url-pattern>/webservice/*</url-pattern>
</servlet-mapping>
```

```
<servlet>
  <servlet-name>HelloPojoService</servlet-name>
  <servlet-class>org.superbiz.servlet.HelloPojoService</servlet-class>
</servlet>
```

```
<servlet-mapping>
  <servlet-name>HelloPojoService</servlet-name>
  <url-pattern>/hello</url-pattern>
</servlet-mapping>
```

```
<login-config>
  <auth-method>BASIC</auth-method>
</login-config>
```

```
<security-role>
  <role-name>manager</role-name>
</security-role>
```

```
<security-role>
  <role-name>user</role-name>
</security-role>
```

```
<env-entry>
  <env-entry-name>web.xml</env-entry-name>
  <env-entry-type>java.lang.String</env-entry-type>
  <env-entry-value>WebValue</env-entry-value>
</env-entry>
```

```
<resource-ref>
  <res-ref-name>web.xml/Data Source</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
</resource-ref>
```

```
<resource-env-ref>
  <resource-env-ref-name>web.xml/Queue</resource-env-ref-name>
  <resource-env-ref-type>javax.jms.Queue</resource-env-ref-type>
</resource-env-ref>
```

```
<ejb-ref>
  <ejb-ref-name>web.xml/EjbRemote</ejb-ref-name>
  <ejb-ref-type>Session</ejb-ref-type>
  <remote>org.superbiz.servlet.AnnotatedEJBRemote</remote>
</ejb-ref>
```

```
<ejb-local-ref>
  <ejb-ref-name>web.xml/EjLocal</ejb-ref-name>
  <ejb-ref-type>Session</ejb-ref-type>
  <local>org.superbiz.servlet.AnnotatedEJBLocal</local>
</ejb-local-ref>
```

```
<persistence-unit-ref>
  <persistence-unit-ref-name>web.xml/PersistenceUnit</persistence-unit-ref-name>
  <persistence-unit-name>jpa-example</persistence-unit-name>
</persistence-unit-ref>
```

```
<persistence-context-ref>
  <persistence-context-ref-name>web.xml/PersistenceContext</persistence-context-ref-
name>
  <persistence-unit-name>jpa-example</persistence-unit-name>
  <persistence-context-type>Transactional</persistence-context-type>
</persistence-context-ref>
</web-app>
```