

# Webservice Inheritance

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

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

## Item

```
package org.superbiz.inheritance;

import javax.persistence.Entity;
import javax.persistence.GeneratedValue;
import javax.persistence.GenerationType;
import javax.persistence.Id;
import javax.persistence.Inheritance;
import javax.persistence.InheritanceType;
import java.io.Serializable;

@Entity
@Inheritance(strategy = InheritanceType.JOINED)
public class Item implements Serializable {
    @Id
    @GeneratedValue(strategy = GenerationType.AUTO)
    private Long id;
    private String brand;
    private String itemName;
    private double price;

    public Long getId() {
        return id;
    }

    public void setId(Long id) {
        this.id = id;
    }

    public String getBrand() {
        return brand;
    }

    public void setBrand(String brand) {
        this.brand = brand;
    }

    public String getItemName() {
        return itemName;
    }
}
```

```

public void setItemName(String itemName) {
    this.itemName = itemName;
}

public double getPrice() {
    return price;
}

public void setPrice(double price) {
    this.price = price;
}
}

```

## Tower

```

package org.superbiz.inheritance;

import javax.persistence.Entity;

@Entity
public class Tower extends Item {
    private Fit fit;
    private String tubing;

    public static enum Fit {
        Custom, Exact, Universal
    }

    public Fit getFit() {
        return fit;
    }

    public void setFit(Fit fit) {
        this.fit = fit;
    }

    public String getTubing() {
        return tubing;
    }

    public void setTubing(String tubing) {
        this.tubing = tubing;
    }

    ;
}

```

# Wakeboard

```
package org.superbiz.inheritance;  
  
import javax.persistence.Entity;  
  
@Entity  
public class Wakeboard extends Wearable {  
}
```

# WakeboardBinding

```
package org.superbiz.inheritance;  
  
import javax.persistence.Entity;  
  
@Entity  
public class WakeboardBinding extends Wearable {  
}
```

# WakeRiderImpl

```

package org.superbiz.inheritance;

import javax.ejb.Stateless;
import javax.jws.WebService;
import javax.persistence.EntityManager;
import javax.persistence.PersistenceContext;
import javax.persistence.PersistenceContextType;
import javax.persistence.Query;
import java.util.List;

/**
 * This is an EJB 3 style pojo stateless session bean Every stateless session
 * bean implementation must be annotated using the annotation @Stateless This
 * EJB has a single interface: {@link WakeRiderWs} a webservice interface.
 */
@Stateless
@WebService(
    portName = "InheritancePort",
    serviceName = "InheritanceWsService",
    targetNamespace = "http://superbiz.org/wsd1",
    endpointInterface = "org.superbiz.inheritance.WakeRiderWs")
public class WakeRiderImpl implements WakeRiderWs {

    @PersistenceContext(unitName = "wakeboard-unit", type = PersistenceContextType
        .TRANSACTION)
    private EntityManager entityManager;

    public void addItem(Item item) throws Exception {
        entityManager.persist(item);
    }

    public void deleteMovie(Item item) throws Exception {
        entityManager.remove(item);
    }

    public List<Item> getItems() throws Exception {
        Query query = entityManager.createQuery("SELECT i FROM Item i");
        List<Item> items = query.getResultList();
        return items;
    }
}

```

## WakeRiderWs

```

package org.superbiz.inheritance;

import javax.jws.WebService;
import javax.xml.bind.annotation.XmlSeeAlso;
import java.util.List;

/**
 * This is an EJB 3 webservice interface that uses inheritance.
 */
@WebService(targetNamespace = "http://superbiz.org/wsdl")
@XmlSeeAlso({Wakeboard.class, WakeboardBinding.class, Tower.class})
public interface WakeRiderWs {
    public void addItem(Item item) throws Exception;

    public void deleteMovie(Item item) throws Exception;

    public List<Item> getItems() throws Exception;
}

```

## Wearable

```

package org.superbiz.inheritance;

import javax.persistence.MappedSuperclass;

@MappedSuperclass
public abstract class Wearable extends Item {
    protected String size;

    public String getSize() {
        return size;
    }

    public void setSize(String size) {
        this.size = size;
    }
}

```

## ejb-jar.xml

```
<ejb-jar/>
```

# persistence.xml

```
<persistence xmlns="http://java.sun.com/xml/ns/persistence" version="1.0">

  <persistence-unit name="wakeboard-unit">

    <jta-data-source>wakeBoardDatabase</jta-data-source>
    <non-jta-data-source>wakeBoardDatabaseUnmanaged</non-jta-data-source>

    <class>org.superbiz.inheritance.Item</class>
    <class>org.superbiz.inheritance.Tower</class>
    <class>org.superbiz.inheritance.Wakeboard</class>
    <class>org.superbiz.inheritance.WakeboardBinding</class>
    <class>org.superbiz.inheritance.Wearable</class>

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

  </persistence-unit>
</persistence>
```

# InheritanceTest

```
package org.superbiz.inheritance;

import junit.framework.TestCase;
import org.superbiz.inheritance.Tower.Fit;

import javax.naming.Context;
import javax.naming.InitialContext;
import javax.xml.namespace.QName;
import javax.xml.ws.Service;
import java.net.URL;
import java.util.List;
import java.util.Properties;

public class InheritanceTest extends TestCase {

  //START SNIPPET: setup
  private InitialContext initialContext;

  protected void setUp() throws Exception {

    Properties p = new Properties();
    p.put(Context.INITIAL_CONTEXT_FACTORY,
```

```

"org.apache.openejb.core.LocalInitialContextFactory");
    p.put("wakeBoardDatabase", "new://Resource?type=DataSource");
    p.put("wakeBoardDatabase.JdbcDriver", "org.hsqldb.jdbcDriver");
    p.put("wakeBoardDatabase.JdbcUrl", "jdbc:hsqldb:mem:wakeBoarddb");

    p.put("wakeBoardDatabaseUnmanaged", "new://Resource?type=DataSource");
    p.put("wakeBoardDatabaseUnmanaged.JdbcDriver", "org.hsqldb.jdbcDriver");
    p.put("wakeBoardDatabaseUnmanaged.JdbcUrl", "jdbc:hsqldb:mem:wakeBoarddb");
    p.put("wakeBoardDatabaseUnmanaged.JtaManaged", "false");

    p.put("openejb.embedded.remotable", "true");

    initialContext = new InitialContext(p);
}
//END SNIPPET: setup

/**
 * Create a webservice client using wsdl url
 *
 * @throws Exception
 */
//START SNIPPET: webservice
public void testInheritanceViaWsInterface() throws Exception {
    Service service = Service.create(
        new URL("http://127.0.0.1:4204/WakeRiderImpl?wsdl"),
        new QName("http://superbiz.org/wsdl", "InheritanceWsService"));
    assertNotNull(service);

    WakeRiderWs ws = service.getPort(WakeRiderWs.class);

    Tower tower = createTower();
    Item item = createItem();
    Wakeboard wakeBoard = createWakeBoard();
    WakeboardBinding wakeBoardbinding = createWakeboardBinding();

    ws.addItem(tower);
    ws.addItem(item);
    ws.addItem(wakeBoard);
    ws.addItem(wakeBoardbinding);

    List<Item> returnedItems = ws.getItems();

    assertEquals("testInheritanceViaWsInterface, nb Items", 4, returnedItems.size
());

    //check tower
    assertEquals("testInheritanceViaWsInterface, first Item", returnedItems.get(0
).getClass(), Tower.class);
    tower = (Tower) returnedItems.get(0);
    assertEquals("testInheritanceViaWsInterface, first Item", tower.getBrand(),

```



```

"Tower brand");
    assertEquals("testInheritanceViaWsInterface, first Item", tower.getFit()
.ordinal(), Fit.Custom.ordinal());
    assertEquals("testInheritanceViaWsInterface, first Item", tower.getItemName(),
"Tower item name");
    assertEquals("testInheritanceViaWsInterface, first Item", tower.getPrice(),
1.0d);
    assertEquals("testInheritanceViaWsInterface, first Item", tower.getTubing(),
"Tower tubing");

    //check item
    assertEquals("testInheritanceViaWsInterface, second Item", returnedItems.get(
1).getClass(), Item.class);
    item = (Item) returnedItems.get(1);
    assertEquals("testInheritanceViaWsInterface, second Item", item.getBrand(),
"Item brand");
    assertEquals("testInheritanceViaWsInterface, second Item", item.getItemName(),
"Item name");
    assertEquals("testInheritanceViaWsInterface, second Item", item.getPrice(),
2.0d);

    //check wakeboard
    assertEquals("testInheritanceViaWsInterface, third Item", returnedItems.get(2
).getClass(), Wakeboard.class);
    wakeBoard = (Wakeboard) returnedItems.get(2);
    assertEquals("testInheritanceViaWsInterface, third Item", wakeBoard.getBrand(
), "Wakeboard brand");
    assertEquals("testInheritanceViaWsInterface, third Item", wakeBoard
.getItemName(), "Wakeboard item name");
    assertEquals("testInheritanceViaWsInterface, third Item", wakeBoard.getPrice(
), 3.0d);
    assertEquals("testInheritanceViaWsInterface, third Item", wakeBoard.getSize(),
"WakeBoard size");

    //check wakeboardbinding
    assertEquals("testInheritanceViaWsInterface, fourth Item", returnedItems.get(
3).getClass(), WakeboardBinding.class);
    wakeBoardbinding = (WakeboardBinding) returnedItems.get(3);
    assertEquals("testInheritanceViaWsInterface, fourth Item", wakeBoardbinding
.getBrand(), "Wakeboardbinding brand");
    assertEquals("testInheritanceViaWsInterface, fourth Item", wakeBoardbinding
.getItemName(), "Wakeboardbinding item name");
    assertEquals("testInheritanceViaWsInterface, fourth Item", wakeBoardbinding
.getPrice(), 4.0d);
    assertEquals("testInheritanceViaWsInterface, fourth Item", wakeBoardbinding
.getSize(), "WakeBoardbinding size");
}
//END SNIPPET: webservice

private Tower createTower() {
    Tower tower = new Tower();

```

```

tower.setBrand("Tower brand");
tower.setFit(Fit.Custom);
tower.setItemName("Tower item name");
tower.setPrice(1.0f);
tower.setTubing("Tower tubing");
return tower;
}

private Item createItem() {
    Item item = new Item();
    item.setBrand("Item brand");
    item.setItemName("Item name");
    item.setPrice(2.0f);
    return item;
}

private Wakeboard createWakeBoard() {
    Wakeboard wakeBoard = new Wakeboard();
    wakeBoard.setBrand("Wakeboard brand");
    wakeBoard.setItemName("Wakeboard item name");
    wakeBoard.setPrice(3.0f);
    wakeBoard.setSize("WakeBoard size");
    return wakeBoard;
}

private WakeboardBinding createWakeboardBinding() {
    WakeboardBinding wakeBoardBinding = new WakeboardBinding();
    wakeBoardBinding.setBrand("Wakeboardbinding brand");
    wakeBoardBinding.setItemName("Wakeboardbinding item name");
    wakeBoardBinding.setPrice(4.0f);
    wakeBoardBinding.setSize("WakeBoardbinding size");
    return wakeBoardBinding;
}
}

```

## Running

```
-----
T E S T S
-----
```

```

Running org.superbiz.inheritance.InheritanceTest
Apache OpenEJB 4.0.0-beta-1    build: 20111002-04:06
http://tomee.apache.org/
INFO - openejb.home = /Users/dblevins/examples/webservice-inheritance
INFO - openejb.base = /Users/dblevins/examples/webservice-inheritance
INFO - Configuring Service(id=Default Security Service, type=SecurityService,
provider-id=Default Security Service)
INFO - Configuring Service(id=Default Transaction Manager, type=TransactionManager,

```

```

provider-id=Default Transaction Manager)
INFO - Configuring Service(id=wakeBoardDatabaseUnmanaged, type=Resource, provider-
id=Default JDBC Database)
INFO - Configuring Service(id=wakeBoardDatabase, type=Resource, provider-id=Default
JDBC Database)
INFO - Found EjbModule in classpath: /Users/dblevins/examples/webservice-
inheritance/target/classes
INFO - Beginning load: /Users/dblevins/examples/webservice-inheritance/target/classes
INFO - Configuring enterprise application: /Users/dblevins/examples/webservice-
inheritance/classpath.ear
INFO - Configuring Service(id=Default Stateless Container, type=Container, provider-
id=Default Stateless Container)
INFO - Auto-creating a container for bean WakeRiderImpl: Container(type=STATELESS,
id=Default Stateless Container)
INFO - Configuring PersistenceUnit(name=wakeboard-unit)
INFO - Enterprise application "/Users/dblevins/examples/webservice-
inheritance/classpath.ear" loaded.
INFO - Assembling app: /Users/dblevins/examples/webservice-inheritance/classpath.ear
INFO - PersistenceUnit(name=wakeboard-unit,
provider=org.apache.openjpa.persistence.PersistenceProviderImpl) - provider time 396ms
INFO - Created Ejb(deployment-id=WakeRiderImpl, ejb-name=WakeRiderImpl,
container=Default Stateless Container)
INFO - Started Ejb(deployment-id=WakeRiderImpl, ejb-name=WakeRiderImpl,
container=Default Stateless Container)
INFO - Deployed Application(path=/Users/dblevins/examples/webservice-
inheritance/classpath.ear)
INFO - Initializing network services
INFO - Creating ServerService(id=httpejbd)
INFO - Creating ServerService(id=cxf)
INFO - Creating ServerService(id=admin)
INFO - Creating ServerService(id=ejbd)
INFO - Creating ServerService(id=ejbds)
INFO - Initializing network services
  ** Starting Services **
  NAME                IP           PORT
  httpejbd            127.0.0.1   4204
  admin thread        127.0.0.1   4200
  ejbd                 127.0.0.1   4201
  ejbd                 127.0.0.1   4203
  -----
Ready!
WARN - Found no persistent property in "org.superbiz.inheritance.WakeboardBinding"
WARN - Found no persistent property in "org.superbiz.inheritance.Wakeboard"
WARN - Found no persistent property in "org.superbiz.inheritance.WakeboardBinding"
WARN - Found no persistent property in "org.superbiz.inheritance.Wakeboard"
Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 4.442 sec

Results :

Tests run: 1, Failures: 0, Errors: 0, Skipped: 0

```