REST Example

Example rest-example can be browsed at https://github.com/apache/tomee/tree/master/examples/rest-example

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

CommentDAO

```
package org.superbiz.rest.dao;

import org.superbiz.rest.model.Comment;
import org.superbiz.rest.model.Post;

import javax.ejb.EJB;
import javax.ejb.Stateless;
import java.util.Collections;
import java.util.List;

@Stateless
public class CommentDAO extends DAO {
    @EJB
    private DAO dao;

    public List<Comment> list(long postId) {
        Post post = dao.find(Post.class, postId);
        if (post == null) {
            throw new IllegalArgumentException("post with id " + postId + " not found
");
        }
        return Collections.unmodifiableList(post.getComments());
    }

    public Comment create(String author, String content, long postId) {
        Post post = dao.find(Post.class, postId);
        if (post == null) {
            throw new IllegalArgumentException("post with id " + postId + " not found
");
        }

        Comment comment = new Comment();
        comment.setAuthor(author);
        comment.setContent(content);
        dao.create(comment);
        comment.setPost(post);
        return comment;
    }

    public void delete(long id) {
        dao.delete(Comment.class, id);
    }

    public Comment update(long id, String author, String content) {
        Comment comment = dao.find(Comment.class, id);
        if (comment == null) {
            throw new IllegalArgumentException("comment with id " + id + " not found"
);
        }

        comment.setAuthor(author);
        comment.setContent(content);
        return dao.update(comment);
    }
}
```

DAO

```
package org.superbiz.rest.dao;

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

/**
 * Simply maps the entitymanager.
 * It simplifies refactoring (unitName change) and wraps some logic (limited queries).
 *
 */
@Stateless
public class DAO {
    @PersistenceContext(unitName = "blog")
    private EntityManager em;

    public <E> E create(E e) {
        em.persist(e);
        return e;
    }

    public <E> E update(E e) {
        return em.merge(e);
    }

    public <E> void delete(Class<E> clazz, long id) {
        em.remove(em.find(clazz, id));
    }
```

```java
    public <E> E find(Class<E> clazz, long id) {
        return em.find(clazz, id);
    }

    public <E> List<E> find(Class<E> clazz, String query, int min, int max) {
        return queryRange(em.createQuery(query, clazz), min, max).getResultList();
    }

    public <E> List<E> namedFind(Class<E> clazz, String query, int min, int max) {
        return queryRange(em.createNamedQuery(query, clazz), min, max).getResultList(
);
    }

    private static Query queryRange(Query query, int min, int max) {
        if (max >= 0) {
            query.setMaxResults(max);
        }
        if (min >= 0) {
            query.setFirstResult(min);
        }
        return query;
    }
}
```

PostDAO

```java
package org.superbiz.rest.dao;

import org.superbiz.rest.model.Post;
import org.superbiz.rest.model.User;

import javax.ejb.EJB;
import javax.ejb.Stateless;
import java.util.List;

@Stateless
public class PostDAO {
    @EJB
    private DAO dao;

    public Post create(String title, String content, long userId) {
        User user = dao.find(User.class, userId);
        Post post = new Post();
        post.setTitle(title);
        post.setContent(content);
        post.setUser(user);
        return dao.create(post);
    }
    public Post find(long id) {
        return dao.find(Post.class, id);
    }

    public List<Post> list(int first, int max) {
        return dao.namedFind(Post.class, "post.list", first, max);
    }

    public void delete(long id) {
        dao.delete(Post.class, id);
    }

    public Post update(long id, long userId, String title, String content) {
        User user = dao.find(User.class, userId);
        if (user == null) {
            throw new IllegalArgumentException("user id " + id + " not found");
        }

        Post post = dao.find(Post.class, id);
        if (post == null) {
            throw new IllegalArgumentException("post id " + id + " not found");
        }

        post.setTitle(title);
        post.setContent(content);
        post.setUser(user);
        return dao.update(post);
    }
}
```

UserDAO

```java
package org.superbiz.rest.dao;

import org.superbiz.rest.model.User;

import javax.ejb.EJB;
import javax.ejb.Stateless;
import java.util.List;

@Stateless
public class UserDAO {
    @EJB
    private DAO dao;

    public User create(String name, String pwd, String mail) {
        User user = new User();
        user.setFullname(name);
        user.setPassword(pwd);
        user.setEmail(mail);
        return dao.create(user);
    }

    public List<User> list(int first, int max) {
        return dao.namedFind(User.class, "user.list", first, max);
    }

    public User find(long id) {
        return dao.find(User.class, id);
    }

    public void delete(long id) {
        dao.delete(User.class, id);
    }

    public User update(long id, String name, String pwd, String mail) {
        User user = dao.find(User.class, id);
        if (user == null) {
            throw new IllegalArgumentException("setUser id " + id + " not found");
        }

        user.setFullname(name);
        user.setPassword(pwd);
        user.setEmail(mail);
        return dao.update(user);
    }
}
```

Comment

```
package org.superbiz.rest.model;

import javax.persistence.Entity;
import javax.persistence.JoinColumn;
import javax.persistence.Lob;
import javax.persistence.ManyToOne;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.validation.Valid;
import javax.validation.constraints.NotNull;
import javax.validation.constraints.Size;
import javax.xml.bind.annotation.XmlRootElement;
import javax.xml.bind.annotation.XmlTransient;

@Entity
@NamedQueries({
        @NamedQuery(name = "comment.list", query = "select c from Comment c")
}
```

DatedModel

```
package org.superbiz.rest.model;

import javax.persistence.MappedSuperclass;
import javax.persistence.PrePersist;
import java.util.Date;

@MappedSuperclass
public abstract class DatedModel extends Model {
    private Date created;

    @PrePersist
    public void create() {
        created = new Date();
    }

    public Date getCreated() {
        return created;
    }

    public void setCreated(Date created) {
        this.created = created;
    }
}
```

Model

```
package org.superbiz.rest.model;

import javax.persistence.Access;
import javax.persistence.AccessType;
import javax.persistence.GeneratedValue;
import javax.persistence.Id;
import javax.persistence.MappedSuperclass;
import javax.xml.bind.annotation.XmlAccessType;
import javax.xml.bind.annotation.XmlAccessorType;

@MappedSuperclass
@Access(AccessType.FIELD)
@XmlAccessorType(XmlAccessType.FIELD)
public abstract class Model {

    @Id
    @GeneratedValue
    protected long id;

    public long getId() {
        return id;
    }

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

Post

```java
package org.superbiz.rest.model;

import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.Lob;
import javax.persistence.ManyToOne;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.persistence.OneToMany;
import javax.validation.Valid;
import javax.validation.constraints.NotNull;
import javax.validation.constraints.Size;
import javax.xml.bind.annotation.XmlRootElement;
import java.util.ArrayList;
import java.util.List;

@Entity
@NamedQueries({
        @NamedQuery(name = "post.list", query = "select p from Post p")
}
```

User

```java
package org.superbiz.rest.model;

import javax.persistence.Entity;
import javax.persistence.NamedQueries;
import javax.persistence.NamedQuery;
import javax.validation.constraints.NotNull;
import javax.validation.constraints.Pattern;
import javax.validation.constraints.Size;
import javax.xml.bind.annotation.XmlRootElement;

@Entity
@NamedQueries({
        @NamedQuery(name = "user.list", query = "select u from User u")
}
```

CommentService

```java
package org.superbiz.rest.service;

import org.superbiz.rest.dao.CommentDAO;
import org.superbiz.rest.model.Comment;
import javax.ejb.EJB;
import javax.ws.rs.DELETE;
import javax.ws.rs.GET;
import javax.ws.rs.POST;
import javax.ws.rs.PUT;
import javax.ws.rs.Path;
import javax.ws.rs.PathParam;
import javax.ws.rs.Produces;
import javax.ws.rs.QueryParam;
import java.util.List;

@Path("/api/comment")
@Produces({"text/xml", "application/json"})
public class CommentService {
    @EJB
    private CommentDAO commentDao;

    @Path("/create")
    @PUT
    public Comment create(@QueryParam("author") String author,
                          @QueryParam("content") String content,
                          @QueryParam("postId") long postId) {
        return commentDao.create(author, content, postId);
    }

    @Path("/list/{postId}")
    @GET
    public List<Comment> list(@PathParam("postId") long postId) {
        return commentDao.list(postId);
    }

    @Path("/delete/{id}")
    @DELETE
    public void delete(@PathParam("id") long id) {
        commentDao.delete(id);
    }

    @Path("/update/{id}")
    @POST
    public Comment update(@PathParam("id") long id,
                          @QueryParam("author") String author,
                          @QueryParam("content") String content) {
        return commentDao.update(id, author, content);
    }
}
```

PostService

```java
package org.superbiz.rest.service;
```

```java
import org.superbiz.rest.dao.PostDAO;
import org.superbiz.rest.model.Post;

import javax.ejb.EJB;
import javax.ws.rs.DELETE;
import javax.ws.rs.DefaultValue;
import javax.ws.rs.GET;
import javax.ws.rs.POST;
import javax.ws.rs.PUT;
import javax.ws.rs.Path;
import javax.ws.rs.PathParam;
import javax.ws.rs.Produces;
import javax.ws.rs.QueryParam;
import java.util.List;

@Path("/api/post")
@Produces({"text/xml", "application/json"})
public class PostService {
    @EJB
    private PostDAO dao;

    @Path("/create")
    @PUT
    public Post create(@QueryParam("title") String title,
                       @QueryParam("content") String content,
                       @QueryParam("userId") long userId) {
        return dao.create(title, content, userId);
    }

    @Path("/list")
    @GET
    public List<Post> list(@QueryParam("first") @DefaultValue("0") int first,
                           @QueryParam("max") @DefaultValue("20") int max) {
        return dao.list(first, max);
    }

    @Path("/show/{id}")
    @GET
    public Post show(@PathParam("id") long id) {
        return dao.find(id);
    }

    @Path("/delete/{id}")
    @DELETE
    public void delete(@PathParam("id") long id) {
        dao.delete(id);
    }

    @Path("/update/{id}")
    @POST
    public Post update(@PathParam("id") long id,
                       @QueryParam("userId") long userId,
                       @QueryParam("title") String title,
                       @QueryParam("content") String content) {
        return dao.update(id, userId, title, content);
    }
}
```

UserService

```
package org.superbiz.rest.service;

import org.superbiz.rest.dao.UserDAO;
import org.superbiz.rest.model.User;

import javax.ejb.EJB;
import javax.ws.rs.DELETE;
import javax.ws.rs.DefaultValue;
import javax.ws.rs.GET;
import javax.ws.rs.POST;
import javax.ws.rs.PUT;
import javax.ws.rs.Path;
import javax.ws.rs.PathParam;
import javax.ws.rs.Produces;
import javax.ws.rs.QueryParam;
import java.util.List;

@Path("/api/user")
@Produces({"text/xml", "application/json"})
public class UserService {
    @EJB
    private UserDAO dao;

    @Path("/create")
    @PUT
    public User create(@QueryParam("name") String name,
                       @QueryParam("pwd") String pwd,
                       @QueryParam("mail") String mail) {
        return dao.create(name, pwd, mail);
    }

    @Path("/list")
    @GET
    public List<User> list(@QueryParam("first") @DefaultValue("0") int first,
                           @QueryParam("max") @DefaultValue("20") int max) {
        return dao.list(first, max);
    }

    @Path("/show/{id}")
```

```java
    @GET
    public User show(@PathParam("id") long id) {
        return dao.find(id);
    }

    @Path("/delete/{id}")
    @DELETE
    public void delete(@PathParam("id") long id) {
        dao.delete(id);
    }

    @Path("/update/{id}")
    @POST
    public User update(@PathParam("id") long id,
                       @QueryParam("name") String name,
                       @QueryParam("pwd") String pwd,
                       @QueryParam("mail") String mail) {
        return dao.update(id, name, pwd, mail);
    }
}
```

persistence.xml

```xml
<persistence version="2.0"
             xmlns="http://java.sun.com/xml/ns/persistence"
             xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
             xsi:schemaLocation="http://java.sun.com/xml/ns/persistence
                       http://java.sun.com/xml/ns/persistence/persistence_2_0.xsd">
  <persistence-unit name="blog">
    <jta-data-source>My DataSource</jta-data-source>
    <non-jta-data-source>My Unmanaged DataSource</non-jta-data-source>
    <class>org.superbiz.rest.model.User</class>
    <class>org.superbiz.rest.model.Post</class>
    <class>org.superbiz.rest.model.Comment</class>
    <class>org.superbiz.rest.model.Model</class>
    <class>org.superbiz.rest.model.DatedModel</class>
    <properties>
      <property name="openjpa.jdbc.SynchronizeMappings" value=
"buildSchema(ForeignKeys=true)"/>
    </properties>
  </persistence-unit>
</persistence>
```

web.xml

```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 REST Example</display-name>
</web-app>
```

UserDaoTest

```
packagenull
}
```

UserServiceTest

```
packagenull
}
```