Java Custom Contest And Submission Entities

1. Design

This component provides Hibernate mapping and object model for contest and studio submission database schema. It is used by Contest Manager and Submission Manager.

There are interface diagram and database schema in development distribution. They should be used during developing of component. All changes should be coordinated with project architect.

This is a development only component.

Version: 1.0

1.1 Design Patterns

None

1.2 Industry Standards

None

1.3 Required Algorithms

Developer is responsible to create full Hibernate mapping for given database schema. Id of entities should be generated by using Hibernate id generation mechanism. Developer should not care about cascade behaviors when deleting/saving/updating entities. This is responsibility of managers and their API.

1.4 Component Class Overview

Package com.topcoder.service.studio.contest

Class Contest
This is entity for contest table

Class StudioFileType
This is entity for file_type_lu table

Class Document
This is entity for document table.

Class DocumentType
This is entity for document_type_lu table. Currently is is possible two types SPECIFICTION, TEMPLATE.

Class ContestType
This is entity for contest_type_lu table

Class Config
This is entity for config table

Class MimeType
This is entity for mime_type_lu table.

Class FilePath
This is entity for path table.

Class ContestStatus
This is entity for contest_status_lu table.

Class ContestCategory
This is entity for contest_category_lu table.

Class ContestStatusIcon
This is entity for contest_status_icon table.

Package com.topcoder.service.studio.submission

Class Submission
This is entity for submission table

Class SubmissionPayment
This is entity for submission_payments table.

Class PaymentStatus
This is entity for payments_status_lu table. Currently is is possible three statuses PAID, UNPAID, MARKED_FOR_PURCHASE

Class SubmissionType
This is entity for prize_type_lu table. Currently is possible two types INITIAL_CONTEST_SUBMISSION_TYPE and FINAL_SUBMISSION_TYPE

Class SubmissionReview
This is entity for submission_review table.

Class SubmissionStatus
This is entity for submission_status_lu table. Currently is is possible three statuses ACTIVE, DELETED.

Class Prize
This is entity for prize table.

Class PrizeType
This is entity for prize_type_lu table. Currently is possible two types CONTEST and BONUS

Class ReviewStatus
This is entity for payments_status_lu table. Currently is possible three statuses PASSED, FAILED
and CHEATED.

Class ContestResult
This is entity for contest_result table.

1.5 Component Exception Definition

No custom exception should be used.

1.6 Thread Safety

Component is not thread safe, because all its classes are mutable.

2. Environment Requirements

2.1 Environment

Development language: Java1.5
Compile target: Java1.5 and Java 1.6

2.2 Top Coder Software Components

None

2.3 Third Party Components

Hibernate 3.2

http://www.hibernate.org/6.html

3. Installation and Configuration

3.1 Package Name

com.topcoder.service.studio.contest
com.topcoder.service.studio.submission

3.2 Configuration Parameters

None

3.3 Dependencies Configuration

None

4. Usage Notes

4.1 Required steps to test the component

Make sure that the hibernate configuration is available in the class path and is pointing to the the correct database.

4.2 Required steps to use the component

None

4.3 Demo

Demo for one entity class FilePath is given. All other entities can be used in a similar fashion.

```
        Session session = HibernateUtil.getSessionFactory().openSession();
        try {
            session.beginTransaction();
            FilePath entity = new FilePath();
            entity.setModifyDate(new Date());
            entity.setPath("path");

            // save the entity
            session.save(entity);

            // load the persisted object
            FilePath persisted = (FilePath) session.get(FilePath.class,
entity.getFilePathId());

            // update the entity
            entity.setPath("new path");
            session.merge(entity);

            persisted = (FilePath) session.get(FilePath.class,
entity.getFilePathId());

            // delete the entity
            session.delete(entity);
            session.getTransaction().commit();
        } finally {
            session.close();

        }
```

5. Future Enhancements

None.