[TopCoder]

Contest And Submission Entities 1.2 Component Specification

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.

In the version 1.2, Contest is modified. ContestGeneralInfo, ContestSpecifications and ContestMultiRoundInformation, and ContestResource are added to support the multi-round contest.

1.1 Design Patterns

- In the TC contest studio application, this component provides the entities, and hence implements the model part of the MVC pattern.

1.2 Industry Standards

- Hibernate 3.2 or higher

1.3 Required Algorithms

1.3.1 Entity fields validation

The entity's set method will not check the field. The valid range of the entities fields are specified in the corresponding DDL schema.

1.4 Component Class Overview

1.4.1 com.topcoder.service.studio.submission

MilestonePrize
Represents the MilestonePrize entity.
It holds the attributes milestone prize id, create date, amount, number of submissions, etc.
It's mutable and not thread safe.

ContestResult
Represents the entity class for db table contest_result.
Thread Safety: This entity is not thread safe since it is mutable.

PaymentStatus
Represents the entity class for db table payment_status_lu.
Currently the three possible statuses are PAID, UNPAID and MARKED_FOR_PURCHASE.
Thread Safety: This entity is not thread safe since it is mutable.

SubmissionType
Represents the entity class for db table submission_type_lu.
Currently two possible statuses are INITIAL_CONTEST_SUBMISSION_TYPE and FINAL_SUBMISSION_TYPE
Thread Safety: This entity is not thread safe since it is mutable.

PrizeType
Represents the entity class for db table prize_type_lu.
Currently two possible types are CONTEST and BONUS.
Thread Safety: This entity is not thread safe since it is mutable.

ReviewStatus
Represents the entity class for db table payments_status_lu.

Currently three possible statuses are PASSED, FAILED and CHEATED.
Thread Safety: This entity is not thread safe since it is mutable.

Submission
Represents the entity class for db table submission.
Changes for Complex Submission Viewer Assembly - Part 2 -
added artifactCount member variable to represent the number of artifacts in submission.
Thread Safety: This entity is not thread safe since it is mutable.

SubmissionPayment
Represents the entity class for db table submission_payments.
Thread Safety: This entity is not thread safe since it is mutable.

SubmissionReview
Represents the entity class for db table submission_review.
Thread Safety: This entity is not thread safe since it is mutable.

SubmissionStatus
Represents the entity class for db table submission_status_lu.
Currently two possible statuses are ACTIVE and DELETED.
Thread Safety: This entity is not thread safe since it is mutable.

Prize
Represents the entity class for db table prize.
Thread Safety: This entity is not thread safe since it is mutable.

1.4.2 com.topcoder.service.studio.contest

ContestGeneralInfo
Represents the ContestGeneralInfo entity.
It holds the attributes contest general info id, goals, target audience, branding guidelines, etc.
It's mutable and not thread safe.

ContestMultiRoundInformation
Represents the ContestMultiRoundInformation entity.
It holds the attributes contest multi round information id, milestone date, submitters locked between rounds, round one introduction, etc.
It's mutable and not thread safe.

ContestSpecifications
Represents the ContestSpecifications entity.
It holds the attributes contest specifications id, colors,fonts, layout and size, etc.
It's mutable and not thread safe.

ContestResource
Represents the ContestResource entity.
It holds the attributes resource id, and name.
It's mutable and not thread safe.

Contest
Represents the entity class for db table contest.
Added the TC Direct Project Name property for Cockpit Release Assembly for Receipts.
Thread Safety: This entity is not thread safe since it is mutable.
Changes in version 1.2:
New attributes are added to support the multi round contest, please see the attributes in the red color.

ContestRegistration
Represents the entity class for db table contest_registration.
Thread Safety: This entity is not thread safe since it is mutable.

ContestType
Represents the entity class for db table contest_type_lu.
Thread Safety: This entity is not thread safe since it is mutable.

ContestProperty
Represents the entity class for db table contest_property_lu.
Thread Safety: This entity is not thread safe since it is mutable.

ContestPayment
Represents the entity class for db table contest_payment.
Thread Safety: This entity is not thread safe since it is mutable.

Document
Represents the entity class for db table document.
Thread Safety: This entity is not thread safe since it is mutable.

DocumentType
Represents the entity class for db table document_type_lu.
Currently the possible types are SPECIFICTION and TEMPLATE.
Thread Safety: This entity is not thread safe since it is mutable.

MimeType
Represents the entity class for db table mime_type_lu.
Thread Safety: This entity is not thread safe since it is mutable.

Medium
Represents the entity class for db table medium_lu.
Thread Safety: This entity is not thread safe since it is mutable.

FilePath
Represents the entity class for db table path.
A sample code showing the CRUD on this entity using hibernate
Thread Safety: This entity is not thread safe since it is mutable.

ContestChannel
Represents the entity class for db table contest_channel_lu.
Thread Safety: This entity is not thread safe since it is mutable.
Version 1.1: removed name, parentChannelId and fileType fields.

ContestTypeConfig
Represents the entity class for db table contest_type_config.
Thread Safety: This entity is not thread safe since it is mutable.

ContestConfig
Represents the entity class for db table config.
Thread Safety: This entity is not thread safe since it is mutable.

StudioFileType
Represents the entity class for db table file_type_lu.
Thread Safety: This entity is not thread safe since it is mutable.

SimpleProjectContestData
Represents the entity class for contest info for myproject widget.
Changes for My Projects Overhaul Assembly:
Added new field contestType which maps to sql column contest_type_desc.
Added new field createUser which maps to sql column create_user.
Thread Safety: This entity is not thread safe since it is mutable.

ContestChangeHistory
Represents the entity class contains information about changes made to a contest, including the contest ID, the field changed, the username making the change, a transaction ID that groups changes made at the same time, and the old data and new data that represents the change.
Thread Safety: This entity is not thread safe since it is mutable.

Helper
This class provides static utility methods which are used to facilitate the coding or reduce the redundancies.
Thread Safety: This class is thread safe since it is immutable.

ContestStatus
Represents the entity class for db table contest_status_lu.
Thread Safety: This entity is not thread safe since it is mutable.

1.4.3 com.topcoder.service.studio

PaymentType
Represents the entity class for payment type. Currently supported types are: Paypal and TC Purchase order.

1.5 Component Exception Definitions

No custom exceptions are defined in this component.

1.6 Thread Safety

JPA/Hibernate does not require entity classes to be thread safe, and those provided by this component indeed are not thread safe too as they are mutable classes. To be thread safe, the caller should guarantee that the entities must not be modified externally during the create/update/delete/ operations.

2. Environment Requirements

2.1 Environment

- At minimum, Java 5.0 is required for compilation and executing test cases
- Test Cases furthermore must be run against Hibernate
- The component is designed particularly to run against Informix database

2.2 TopCoder Software Components

- **None**.

2.3 Third Party Components

- None.

3. Installation and Configuration

3.1 Package Name

- com.topcoder.service.studio
- com.topcoder.service.studio.submission

- com.topcoder.service.studio.contest

3.2 Configuration Parameters

- None.

3.3 Dependencies Configuration

- None.

4. Usage Notes

4.1 Required steps to test the component

- Extract the component distribution.
- Execute 'ant test' within the directory that the distribution was extracted to.

4.2 Required steps to use the component

The component provides entities and configuration for use with Hibnerate.

4.3 Demo

Please note that the demo in the previous version is dummy. The whole demo is enhanced, the new API and entities have the red color.

4.3.1 The usage of classes as java beans

```
// create the ContestGeneralInfo instance
ContestGeneralInfo contestGeneralInfo = new ContestGeneralInfo();

// we should assume that all the fields set the
// entities were retrieved somewhere.
// e.g. the goals set to contestGeneralInfo was
// initialized somewhere else.

// set goals
contestGeneralInfo.setGoals(goals);

// set target audience
contestGeneralInfo.setTargetAudience(targetAudience);

// set branding guidelines
contestGeneralInfo.setBrandingGuidelines(brandingGuidelines);

// set disliked designs websites
contestGeneralInfo.setDislikedDesignsWebsites(dislikedDesignsWebs
ites);

// set other instructions
contestGeneralInfo.setOtherInstructions(otherInstructions);

// create the ContestSpecifications instance
ContestSpecifications contestSpecifications = new
   ContestSpecifications();

// set colors
contestSpecifications.setColors(colors);

// set fonts
contestSpecifications.setFonts(fonts);

// create the ContestMultiRoundInformation instance
```

```
ContestMultiRoundInformation contestMultiRoundInformation = new
    ContestMultiRoundInformation();

// set milestone date
contestMultiRoundInformation.setMilestoneDate(milestoneDate);

// set round one introduction
contestMultiRoundInformation.setRoundOneIntroduction(roundOneIntr
oduction);

// set submitters locked between rounds
contestMultiRoundInformation.setSubmittersLockedBetweenRounds(sub
mittersLockedBetweenRounds);

// create the Contest instance
Contest contest = new Contest();

// set specifications
contest.setSpecifications(contestSpecifications);

// set multi round information
contest.setMultiRoundInformation(contestMultiRoundInformation);

// set milestone prize, assume that the ilestonePrize was created
contest.setMilestonePrize(milestonePrize);

// set general info
contest.setGeneralInfo(contestGeneralInfo);

// set multi round
contest.setMultiRound(new Boolean(true));

// users can also set the resources for the contest.
// this is not showed, it's similar

// set name
contest.setName(name);

// set contest channel
contest.setContestChannel(contestChannel);

// set project id
contest.setProjectId(projectId);

// set tc direct project id
contest.setTcDirectProjectId(tcDirectProjectId);

// set status
contest.setStatus(status);

// set forum id
contest.setForumId(forumId);

// set event id
contest.setEventId(eventId);

// set submissions
```

```
contest.setSubmissions(submissions);

// set file types
contest.setFileTypes(fileTypes);

// set results
contest.setResults(results);

// set media
contest.setMedia(media);

// set documents
contest.setDocuments(documents);

// set config
contest.setConfig(config);

// set contest type
contest.setContestType(contestType);

// set start date
contest.setStartDate(startDate);

// set end date
contest.setEndDate(endDate);

// set winner annoucement deadline
contest.setWinnerAnnoucementDeadline(winnerAnnoucementDeadline);

// set prizes
contest.setPrizes(prizes);

// set non winning submissions purchased
contest.setNonWinningSubmissionsPurchased(nonWinningSubmissionsPu
rchased);

// set resources
contest.setResources(resources);

// developers are free to set other fields for the contest
```

4.3.2 *The usage of classes as entities via Hibernate*

```
// In this demo, we assume that the entities are initialized
// in the demo 4.3.1

// assume that HibernateUtil was set up
Session session = HibernateUtil.getSessionFactory().openSession();
try {
   session.beginTransaction();
   session.save(contest);

   Contest result = (Contest)
      session.get(Contest.class, contest.getContestId());

   // check the fields
   System.out.println(result.getName());
   System.out.println(result.getForumId());
```

```
        // update the contest name
        result.setName("Contest and Submission entities version 1.2");
        session.merge(result);

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


// NOTE: Manipulations on the other entities are quite
// similar, so they are not shown.
```

5. Future Enhancements

- None.