

Java Custom Studio Service

1. Scope

1.1 Overview

This component provides service to get, update, and create contest data; get, remove and update submission data; get some additional information like content's categories, statuses and file types. It defines web service interface and provides its EJB endpoint implementation. Component uses Contest Manager and Submission Manager for providing all operations with persistence. It should not access to persistence by itself.

Each contest has only one client who is owner of contest project. Only this client can access to contest, submissions, document etc. Administrators have also access to contests and their privileges are higher than client ones.

1.2 Version

1.0

1.2 Logic Requirements

1.2.1 Provide operations on the contest

- Create new contest. Method return contest populated by id. After the creation of actual contest, method should check documents and save if it is needed(file does not exist) to server file system. Any error during file writing should not cause the exception on service level. All kinds of I/O errors should be only logged. Note that as creator user it should be set id of current user either client or administrator. This operation is allowed by client who owns the contest project or for administrator.
- Get contest by id. After getting contest, content of each document should be retrieved and embedded into ContestData. Any error of file reading should not cause the exception on service level. All kinds of I/O errors should be only logged. Administration can get any contest, client can get only contest, which belongs to his/her project

- Get project's contest. After getting contest, content of each document should be retrieved and embedded into ContestData. Any error of file reading should not cause the exception on service level. All kinds of I/O errors should be only logged. Administration can get contests for any project, client can get contests for own project.
- Update contest. Administrator can update any contest, client can update only contest, which belongs to his/her project. After the updating of actual contest, method should check documents and save if it is needed (file does not exist) to server file system. Any error during file writing should not cause the exception on service level. All kinds of I/O errors should be only logged. Note that contest can be fully updated if it is only in draft mode status. If contest is in the other status it is possible to increase only prize and duration in hours. This will be checked by Contest Manager component.
- Update contest status. This is administrative function only. Statuses can be update in strict order. See the Chart Diagram as additions.
- Upload document for contest. Before the creation of actual contest, method should check documents and save if it is needed (file does not exist) to server file system. Administrator can upload document to any contest and in any status. Client can upload document only in draft mode status and only for own contest.
- Remove uploaded document from contest. Administrator can remove document from any contest and in any status. Client can remove document only in draft mode status and only from own contest.

See interface diagram for methods API

1.2.2 *Provide operations on the submissions*

- Retrieve submissions for contest. It is possible to retrieve submissions, which passed screening (attribute passedScreening of ContestResult is true). Client can get either full submissions or submission which contains only watermarked images. Client can get full submission only if he/she paid for it. This is defined by payments status. Administrator always gets full submission. Administrator can get submissions from any contest for any payments status. Client can get submission only from contest which belongs to his/her project

- Retrieve submission for member. This is administrative operation only and administrator always retrieves full submissions.
- Update submission result e.g. update submission rank and add prize for submission depends on rank. Note that under contest only 1 submission can have first place, only 1 can have second place etc. Ties are not allowed. This operation is allowed by client who owns the contest project or for administrator.
- Remove submission. Status of submission should be "deleted". The submission is not actually deleted from the system. Note that removed submissions are invisible for select operations. This operation is allowed by client who owns the contest project or for administrator.

1.2.3 *Provide utility operations*

Service also provides some useful utility operations. These operations are acceptable for all clients.

- Retrieve all contest categories
- Retrieve all contest statuses
- Retrieve all contest file types

See interface diagram for methods API

1.2.4 *Service description*

Created service should correspond given WSDL description. Any change or update of WSDL should be confirmed by PM. Custom exception should correspond fault messages defined in WSDL.

1.2.5 *EJB security description*

All its methods should have security role access - "User", except methods which gets all prerequisite documents. Its role should be "Administrator". Note that security identity of EJB should be also "Administrator".

1.2.6 *Logging*

All defined operations should be logged. The logging mechanism should be pluggable e.g. it should be option to disable logging. Logging strategy:

- Entrance and exit of methods should be logged at the INFO level
- Exception should be logged at the ERROR level

1.2.7 Explanation of some ContestData's attributes

launchDateAndTime	Actually this is start date in terms of Contest
durationInHours	This is Contest.startDate - Contest.endDate in hours
finalFileFormat	Default file format(extension) for contest. It gets from contest category
otherFileFormats	All possible formats(extensions) available for contest separated by commas
contestPayload	This is configuration parameters of content type. They should be never added through this service, they can be only retrieved.
contestDescriptionAndRequirements	Contest configuration parameter
requiredOrRestrictedColors	Contest configuration parameter
requiredOrRestrictedFonts	Contest configuration parameter
sizeRequirements	Contest configuration parameter
otherRequirementsOrRestrictions	Contest configuration parameter

1.2.8 Get role and identifier of current user

- Get user principal by calling getCallerPrincipal() method on the EJBContext
- It should be returned UserProfilePrincipal instance defined in JBoss Login Module
- User roles will be found in the "roles" attribute of the Profile instance - role id and values are strings - role names.
- User id should be stored as attribute of UserProfilePrincipal

Also to get client of contest or project is possible by using corresponding methods of ContentManager. Note that as project to define client should be used tcDirectProjectId attribute of Contest.

1.3 Required Algorithms

None.

1.4 Example of the Software Usage

This component will be used to provide possibilities to get, update, and create contest data; get, remove and update submission data; get some additional information like content's categories, statuses and file types.

1.5 Future Component Direction

None.

2. Interface Requirements

2.1.1 Graphical User Interface Requirements

None

2.1.2 External Interfaces



See Interface Diagram:

2.1.3 Environment Requirements

- Development language: Java 1.5
- Compile target: Java 1.5, Java 1.6

2.1.4 Package Structure

com.topcoder.service.studio

3. Software Requirements

3.1 Administration Requirements

3.1.1 What elements of the application need to be configurable?

Logging should be pluggable

3.2 Technical Constraints

3.2.1 Are there particular frameworks or standards that are required?

- EJB 3.0
- WSDL 1.1

3.2.2 TopCoder Software Component Dependencies:

- Base Exception 2.0
- Logging Wrapper 2.0
- Contest Manager 1.0
- Submission Manager 1.0
- JBoss Login Module 2.0

**Please review the TopCoder Software component catalog for existing components that can be used in the design.

3.2.3 Third Party Component, Library, or Product Dependencies:

- None.

3.2.4 QA Environment:

- RedHat Linux 9
- Informix 10
- JBoss 4.2

3.3 Design Constraints

The component design and development solutions must adhere to the guidelines as outlined in the TopCoder Software Component Guidelines. Modifications to these guidelines for this component should be detailed below.

3.4 Required Documentation

3.4.1 Design Documentation

- Use-Case Diagram
- Class Diagram
- Sequence Diagram
- Component Specification

3.4.2 Help / User Documentation

- Design documents must clearly define intended component usage in the 'Documentation' tab of Poseidon.