[TopCoder]

Contest_Eligibility_Persistence_1.0 Component Specification

1. Design

As TopCoder expands and adds more clients, we want to be able to open up more flexibility to customers in the way that they are able to run contests. One such flexibility is to be able to set eligibility conditions on a contest. For example, suppose Client A only wants to allow members from group X to participate. Or, only allow members from certain schools to participate. The types and conditions would be flexible.

This component will provide contest eligibility entities and basic persistence. This is a dev only component.

This component provides the interface and EJB implementation to manage the contest eligibility entities in the persistence.

1.1 Design Patterns

Strategy pattern – this component provides contest eligibility entity management interface and one its EJB implementation.

DAO pattern – this component provides ContestEligibilityManager interface that are DAO for entities.

Business Interface – When hosted in EJB container, the bean can be accessed by the local interface.

1.2 Industry Standards

EJB, IoC, JPA, JPQL,INFORMIX

1.3 Required Algorithms

1.3.1 The save list operations

Save a list of eligibilities, if can be create/update/delete.
You will get a list of eligibilities as input parameter, for each eligibility in the list, you do one of these
for (ContestEligibility contestEligibility : list) {
create/insert, if id is 0 then insert.
update, if id is not 0, do an update.
delete, if 'delete' flag is true, you remove the eligibility.
}
Note remove the deleted contestEligibility entity in return list.

1.3.2 Logging

1. The method entrance/exit will be logged as DEBUG level, including the method name.

2. The parameters will be logged at INFO level.

3. The error will be logged at ERROR level before thrown. The exception error message should be logged.

All other algorithms are simple and straight-forward.

1.4 Component Class Overview

1.4.1 com.topcoder.service.contest.eligibility

ContestEligibility

Represents the base class for contest eligibility.

The entity is not intended to be thread safe, as they only carry data between the layers of an application.

Note that the same contract related to thread-safety applies to all the sub-classes.

GroupContestEligibility

Represents the group contest eligibility entity.

Thread Safety: This entity is not thread safe since it is mutable.

1.4.2 com.topcoder.service.contest.eligibility.dao

ContestEligibilityManager [interface]
This is a generic interface that defines methods that are for all contest eligibility entity management services provided in this component.

ContestEligibilityManagerLocal [interface]
This interface represents the local interface for ContestEligibilityManagerBean session bean. It extends ContestEligibilityManager interface and provides no additional methods.

ContestEligibilityManagerRemote [interface]
This interface represents the remote interface for ContestEligibilityManagerBean session bean. It extends ContestEligibilityManager interface and provides no additional methods.

ContestEligibilityManagerBean
This class is a stateless session bean implementation of the ContestEligibilityManager business interface. It extends ContestEligibilityManagerLocal class and ContestEligibilityManagerRemote class.

1.5 Component Exception Definitions

ContestEligibilityPersistenceException
This exception is thrown by implementations of ContestEligibilityManager when error occurs in the persistence layer.

1.6 System Exception

IllegalArgumentException

This exception is thrown if given parameter is illegal.

1.6 Thread Safety

Implementation of ContestEligibilityManager interface must be thread safe when entities passed to them are used in thread safe manner by the caller.

EJB bean provided in this component use container managed transactions. Their methods that change the database data are annotated to indicate that the transaction is required. And reading methods are annotated to indicate that transactions are allowed.

Transactions are rolled back automatically when ContestEligibilityPersistenceException is thrown (@ApplicationException(rollback=true) annotation is used for this).

2. Environment Requirements

2.1 Environment

2.1.1 Particular frameworks or standards that are required

Development language: Java1.5, J2EE 1.5
Compile target: Java1.5, J2EE 1.5
The component is designed particularly to run against Informix database

2.2 TopCoder Software Components

Base Exception 2.0 – is used for Logging Wrapper.

Logging Wrapper 2.0 – is used for logging errors and debug information.

Object Formatter 1.0 – is used for Logging Wrapper

Typesafe Enum.jar 1.0 – is used for Logging Wrapper

NOTE: The default location for TopCoder Software component jars is../lib/tcs/COMPONENT_NAME/COMPONENT_VERSION relative to the component installation. Setting the tcs_libdir property in topcoder_global.properties will overwrite this default location.

2.3 Third Party Components

Hibernate (https://www.hibernate.org/) as implementation of JPA.

Informix JDBC Driver

NOTE: The default location for 3rd party packages is ../lib relative to this component installation. Setting the ext_libdir property in topcoder_global.properties will overwrite this default location.

3. Installation and Configuration

3.1 Package Name

com.topcoder.service.contest.eligibility
com.topcoder.service.contest.eligibility.dao

3.2 Configuration Parameters

3.2.1 EJB3 configuration of ContestEligibilityManagerBean

Parameter name	Value	Required	Default
unitName	Name of persistence unit. Not null and not empty.	true	N/A
logName	Name of log. Could not be null.Can be empty if you really want	false	contest_eligibility_logger

EJB deployment file is:

```
<ejb-jar 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/ejb-jar_3_0.xsd"
    version="3.0">
    <enterprise-beans>
        <session>
            <ejb-name>contest_eligibility_persistence</ejb-name>

<remote>com.topcoder.service.contest.eligibility.dao.ContestEligibilityManagerRe
mote</remote>

<local>com.topcoder.service.contest.eligibility.dao.ContestEligibilityManagerLoc
al</local>
            <ejb-
class>com.topcoder.service.contest.eligibility.dao.ContestEligibilityManagerBean
</ejb-class>
            <session-type>Stateless</session-type>
            <transaction-type>Container</transaction-type>
            <env-entry>
                <env-entry-name>logName</env-entry-name>
```

```
                <env-entry-type>java.lang.String</env-entry-type>
                <env-entry-value>contest_eligibility_persistence_logger</env-
entry-value>
            </env-entry>
            <persistence-context-ref>
                <persistence-context-ref-name>unitName</persistence-context-ref-
name>
                <persistence-unit-name>persistence-unit</persistence-unit-name>
            </persistence-context-ref>
        </session>
    </enterprise-beans>
</ejb-jar>
```

3.3 Dependencies Configuration

Please refer to 3.2

4. Usage Notes

4.1 Required steps to test the component

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

4.2 Required steps to use the component

Please see the demo.

4.3 Demo

4.3.1 ContestEligibilityManager usage

```
// Acquire a ContestEligibilityManager
1  EJB get local injection @EJB
2  JNDI get remote injection
  Context context = new InitialContext();
  ContestEligibilityManager contestEligibilityManager =
              (ContestEligibilityManager) context
               .lookup("contest_eligibility_persistence/ContestEligibilityManag
erBean/remote");
// create a GroupContestEligibility instance named groupContestEligibility
     ContestEligibility groupContestEligibility = new GroupContestEligibility();
     groupContestEligibility.setContestId(16);
     groupContestEligibility.setStudio(true);

        // insert groupContestEligibility into DB
        groupContestEligibility =
contestEligibilityManager.create(groupContestEligibility);

        // get a list of eligibilities for a contest
        List<ContestEligibility> list =
contestEligibilityManager.getContestEligibility(16, true);
        System.out.println(list.size());

        // Save a list of eligibilities,you can add/update/delete entities.Here
we just update it. You also
        // can refer to ContestEligibilityManagerBeanTests for more tests to
add/update/delete.
        contestEligibilityManager.save(list);
```

```
// Remove a contest eligibility
      contestEligibilityManager.remove(groupContestEligibility);
```

5. Future Enhancements

Additional eligibility types will be added.