Software Documentation : Java Custom Project Payment Management

This page last changed on Jan 23, 2013 by volodymyrk.

1. Scope

1.1 Overview

The Project Payment Management component will define new entities for project payments and provide basic CRUD operations for them.

1.2 Logic Requirements

1.2.1 Project Payment entity

Project Payment is a new entity that describes a single payment assigned to a resource within a project (e.g. review payment, submitter's prize payment, copilot payment etc.). It will contain the following fields:

1. ID.
2. Project Payment Type (see section 1.2.2). Can't be null.
3. Resource ID. This is the ID of the resource this payment is assigned to. Can't be null.
4. Submission ID. This is the ID of the submission that this payment is for. Can be null (e.g. for review payments which are not associated with any particular submission).
5. Amount. Can't be null or negative. Use BigDecimal (with scale 2 and rounding mode HALF_UP) to represent monetary values.
6. PACTS payment ID. Will be null if the payment is not yet transferred to PACTS, otherwise it will contain the ID of the payment in PACTS.
7. Create Date. Can't be null.

The records will be stored in the tcs_catalog.project_payment table. The component will provide an entity class for the project payment. This class will include all the fields from the corresponding table. The entity class will not perform any fields validation, this will be done by the manager classes where applicable.

The DDL for the table is provided within the DB schema.

1.2.2 Project Payment Type entity

Project Payment Type is a new entity that describes the type of the project payments. It will contain the following fields:

1. ID.
2. Name. Can't be null or empty.
3. Mergeable flag. Determines whether multiple payments of this type for the same resource can be merged when transferring to PACTS.

The records will be stored in the tcs_catalog.project_payment_type_lu table. The component will provide an entity class for the project payment type. This class will include all the fields from the corresponding table. The entity class will not perform any fields validation, this will be done by the manager classes where applicable.

The DDL for the table is provided within the DB schema.

1.2.3 Project Payment Manager

The component will provide a manager interface and implementation for the Project Payment entity that will provide the creating, updating, retrieving, deleting and searching functionality.

1.2.3.1 Create operation

The component will provide a method to create a new Project Payment entry. The method will perform validation of the entry field values.

The new entry ID will be generated by the DB sequence. The method will return the newly created entry with set ID value.

1.2.3.2 Update operation

The component will provide a method to update an existing Project Payment entry. The method will perform validation of the entry field values.

1.2.3.3 Retrieve operation

The component will provide a method to retrieve an existing Project Payment entry by its ID. If the entry with the given ID doesn't exist the method will return null.

1.2.3.4 Delete operation

The component will provide a method to delete an existing Project Payment entry by its ID. The method's return value will indicate if the entry with the given ID was deleted.

1.2.3.5 Search operation

The component will provide a method for searching Project Payment entries. The methods will take Filter object which defines the search constraints for the project payments. Search Builder component will be used to work with the search filters. It is the responsibility of the designer to write the search bundle's context and configuration files.

The component will also provide a static helper class that provides factory methods for creating the following filters :

1. Filter by resource ID.
2. Filter by project ID. The search bundle context needs to join the tcs_catalog.resource table to be able to filter by the project ID.
3. Filter by the project payment type ID.
4. Filter by submission ID.
5. Filter by whether the payment has been transferred to PACTS (this is a boolean filter), i.e. whether the pacts_payment_id is not null.

1.2.4 Project Payment Adjustment entity

Project Payment Adjustment is a new entity that describes an adjustment to a default payment which is to be assigned to a resource. For example, it can describe that the default review payment for a particular project needs to be raised by 50%, or to a fixed amount. The entity will contain the following fields:

1. Project ID. Can't be null. This is the ID of the project for which the default payment is to be adjusted.
2. Resource role ID. Can't be null. This is the ID of the resource role for which the default payment is to be adjusted.
3. Fixed amount. Can be null but can't be negative. Use BigDecimal (with scale 2 and rounding mode HALF_UP) to represent monetary values. If this value is not null it describes the fixed amount that should be paid for this project and resource role.
4. Multiplier. Can be null but can't be negative. If this value is not null, it is multiplied by the default payment value to get the new payment amount for this project and resource role.

If both fixed amount and multiplier fields are null, it means that the default payment value will be applied for this project and resource role. Either fixed amount or multiplies has to be null (i.e. they can't take non-null values at the same time).

The records will be stored in the tcs_catalog.project_payment_adjustment table. The component will provide an entity class for the project payment type. This class will include all the fields from the corresponding table. The entity class will not perform any fields validation, this will be done by the manager classes where applicable.

The DDL for the table is provided within the DB schema.

1.2.5 Project Payment Adjustment Manager

The component will provide a manager interface and implementation for the Project Payment Adjustment entity that will provide the creating/updating and retrieving functionality.

1.2.5.1 Create/Update operation

The component will provide a method to create/update a new Project Payment Adjustment entry. The method will perform validation of the entry field values.

The method will update the existing record in the DB, or if it doesn't exist yet it will create a new entry. There are no separate methods to create and update an entry because a missing entry means the default payment, so creating an entry can be treated as an update operation.

1.2.5.2 Retrieve operation

The component will provide a method to retrieve all Project Payment Adjustment records for a project ID.

1.2.6 Thread-Safety

The component is required to be thread-safe.

1.2.7 Error handling

The component should define a strategy for handling all types of errors that can happen in the operations above. This includes validation errors, updating a non-existent entry, any DB-related error etc.

1.3 Required Algorithms

None.

1.4 Example of the Software Usage

The component will be used by the Online Review application to manage the project payment and payment adjustment entries.

1.5 Future Component Direction

Any enhancement needs to be approved either in forum or in email with managers to eliminate over-complicating the component with useless functions. All performance optimizations are highly encouraged and do not require explicit approval.

2. Interface Requirements

2.1.1 Graphical User Interface Requirements

None, only API interface will be provided.

2.1.2 External Interfaces

None.

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.management.payment

3. Software Requirements

3.1 Administration Requirements

3.1.1 What elements of the application need to be configurable?

None.

3.2 Technical Constraints

3.2.1 Are there particular frameworks or standards that are required?

None.

3.2.2 TopCoder Software Component Dependencies:

- Base Exception 2.0.0
- Configuration API 1.1.0
- Configuration Persistence 1.0.2
- DB Connection Factory 1.1.1
- Logging Wrapper 2.0.0
- Search Builder 1.3.2

**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:

Any third party library needs to be approved.

3.2.4 QA Environment:

- Java 1.5
- RedHat Linux 4
- Windows 2000
- Windows 2003

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
- Configuration Files for the Search Builder component

3.4.2 Help / User Documentation

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