This page last changed on Aug 24, 2008 by ghostar.

# 1. Scope

## 1.1 Overview

This component provides a wrapper to the Client and Project Management component, exposing the main management functionality as webservices. This component must also provide client classes that will allow for easy remote access to the service methods.  This component addresses the Client, Project, and Company management services, as well as the authorization pieces associated with those services.  The lookup and status management services are part of a separate component.

### 1.1.1 Version

1.0

## 1.2 Logic Requirements

### 1.2.1 Interfaces

The interfaces for this component can be found in the provided tcuml file, in the "Web Services" class diagram. Both Local and Remote EJB3 interfaces are required.

### 1.2.2 EJB Security

All service methods should have a security role named "User" applied. The methods will use the caller principal name when updating or creating, to set the "modifyUsername" property of the entity(ies) being updated, along with the current date and time to set the "modifyDate" property.

JBoss Login Module 2.0 must be used to retrieve the principal information in the service calls.

### 1.2.3 User Mapping Retreiver

The UserMappingRetriever retrieves information about the clients and projects a specific user ID has access to. This interface is used by the services to ensure that a caller has access to the information about a specific client or project before returning the data. The default implementation should use the xref tables in the database and the entity classes defined to retrieve the user information.

### 1.2.4 Service Definitions

Due to the requirements of the Client and Project Services application, the retrieval of information is separate to a different webservice, with enhanced security. When making calls to the LookupService, a user must be in the "Client and Projects User" role. Not only that, but when retrieving projects and client information, the service must ensure that the user ID making the call is part of the list returned from a call to UserMappingRetriever.getValidUsersForClient or getValidUsersForProject. This will ensure that one client can't retrieve information for a different client's projects.

This same approach should be taken when calls are made to the project and client manipulation methods in the ProjectService and ClientServices as well, to ensure a client can only update it's own record and projects. A project should never be allowed to be created and associated with a client other than what the calling user has access to.

### 1.2.5 Logging

Each service method called should be logged at the DEBUG level, including the parameter information. If errors occur in any call, a message should be logged at the WARNING level before the exception is re-thrown. The logging will be done through Logging Wrapper 2.0

### 1.2.6 Entities

The entities will have proper XML serialization attributes applied to them. The entities are part of the Client and Project Entities and DAO component.

### 1.2.7 Webservice Client

This component must define a client class that mimics the API of the service, to make it easy for a user to access the webservices remotely.

## 1.3 Required Algorithms

None.

## 1.4 Example of the Software Usage

This component will be used to access the Client and Projects Management component remotely.

## 1.5 Future Component Direction

If the Client and Projects Management component is updated, this component will be updated to reflect the changes.

# 2. Interface Requirements

### 2.1.1 Graphical User Interface Requirements

None.

### 2.1.2 External Interfaces

Designs must adhere to the interface diagram definition found in the architecture TCUML file provided. Changes to the interfaces should be approved in the forum.

This component will include the classes detailed below, as well as:

- UserMappingRetriever
- UserMappingRetrievalException
- UserClientMapping
- UserProjectMapping
- JPAUserMappingRetriever

The following services are part of this component:

- ProjectService
- ClientService
- CompanyService

This will include the following dependent interfaces:

- ProjectServiceRemote / ProjectServiceLocal
- ClientServiceRemote / ClientServiceLocal

- CompanyServiceRemote / CompanyServiceLocal
- ProjectServiceBean
- ClientServiceBean
- CompanyServiceBean
- ProjectServiceClient
- ClientServiceClient
- CompanyServiceClient

The base exceptions shown are also part of this component.

### 2.1.3 Environment Requirements

- Development language: Java1.5
- Compile target: Java1.5 and Java1.6

### 2.1.4 Package Structure

com.topcoder.clients.webservices
com.topcoder.clients.webservices.bean
com.topcoder.clients.webservices.webserviceclients

# 3. Software Requirements

## 3.1 Administration Requirements

### 3.1.1 What elements of the application need to be configurable?

- The manager interface instances to use.

## 3.2 Technical Constraints

### 3.2.1 Are there particular frameworks or standards that are required?

None.

### 3.2.2 TopCoder Software Component Dependencies:

- Search Builder 1.4
- Contest and Project Entities and DAO 1.0
- Contest and Project Management 1.0
- Base Exception 2.0
- JBoss Login Module 2.0
- Logging Wrapper 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:

- Solaris 7
- RedHat Linux 7.1
- Windows 2000
- Windows 2003
- Informix 10

## 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

None

### 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 the TopCoder UML Tool.