



Software Documentation : Java Custom Game Plan Service

This page last changed on May 21, 2010 by [hohosky](#).

1. Scope

1.1 Overview

The Game Plan Service component defines the interface to retrieve and aggregate data related to the game plan of TopCoder Direct Projects and DTOs which encapsulate the retrieved data. The component also provides a EJB3 implementation of the defined interface.

This component will be used by new TopCoder Direct site to implement the Game Plan related features.

1.1.1 Version

1.0

1.2 Logic Requirements

1.2.1 Interface

This component shall define and implement the interface for Game Plan Service to meet the following requirements. Note that all the service methods should take a TCSUBJECT as input.

Source of TCSUBJECT class is available at [security.zip](#)

1.2.2 Retrieve game plan related data

The component shall retrieve the Game Plan related data for all the projects (i.e. software projects and studio projects) under TC Direct Projects.

If the user is in Role admin, all the active TC Direct Projects should be retrieved. If the user is not in Role admin, only TC Direct projects belong to current user are retrieved.

For each TC Direct Project, the following data shall be retrieved:

1.2.2.1 Software Projects under TC Direct Project

For each TC Direct project, the component shall retrieve the following data of each software project under the TC Direct Project from data source *tcs_catalog*:

Note that specification review project should be excluded.

- TC direct project ID
 - Project ID
 - Project name
 - Project start date and end date
 - Create user
 - Project dependencies (i.e. project IDs this project depends on)
- There are 5 project link types defined in *link_type_lu* table:

link_type_id	link_type_name	allow_overlap
1	Depends On	0
2	Is Related To	1
3	Requires Spec Review	0
4	For Design	0
5	Repost For	1

Only link types 1 and 4 are counted as project dependencies



- Whether project is a repost (check project links of link type 5 'Repost For')
- Project status
- Project type (e.g. Conceptualization, Specification, Architecture, Component Design etc.)
- Whether project is started

1.2.2.2 Studio projects under TC Direct Project

For each TC Direct project, the component shall retrieve the following data of each studio project under the TC Direct Project from data source *studio_oltp*:

- TC direct project ID
- Project ID
- Contest ID
- Project name
- Contest Name
- Start date and end date
- Create user
- Contest status
- Contest type (e.g. Conceptualization, Specification, Architecture, Component Design etc.)
- Whether contest is started

1.2.3 DTOs

DTOs should be defined to encapsulate the data mentioned in 1.2.2. Please put the DTOs under component Cockpit Facade Util, so they can be used in other services too.

1.2.4 Authentication

No authentication should be performed in this component. Authentication will be done in an upper layer.

1.2.5 Return values

For all methods/services returning list or array, in case there are no objects to fill the list / array, must return an empty one, not null.

1.2.6 Access to data persistence

We do have EJB services implemented to retrieve the similar data from persistence, but these services also retrieve a lot of unrelated data that makes the call slow. So we'd like directly query the persistence to retrieve the Game Plan related data. The following components will be provided for reference but will not be used:

- Project Services: EJB service for software project related stuff.
- Project Management: used by Project Services.
- Project Management Persistence: used by Project Management component to do persistence.
- Studio Service: EJB Service for studio contests
- Studio Contest Manager: Used by studio service
- Contest Service Facade: Facade on Project Services and Studio Service

This component will retrieve data with two queries to two data sources (one for studio, one for software), check method `ContestServiceFacadeBean#getCommonProjectContestData` in component Contest Service Facade to get an idea.

1.2.7 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

1.3 Required Algorithms

None.

1.4 Example of the Software Usage

The new TopCoder Direct site will use this component to implement Game Plan features, the retrieved data will be used to render a Gantt chart which shows the game plan of project.

1.5 Future Component Direction

New services will be added to support the save of the game plan.

2. Interface Requirements

2.1.1 Graphical User Interface Requirements

None.

2.1.2 External Interfaces

2.1.3 Environment Requirements

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

2.1.4 Package Structure

com.topcoder.service.gameplan

com.topcoder.service.gameplan.ejb

com.topcoder.service.util.gameplan - put DTOs under this package in component *Cockpit Facade Util*

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?

- Java 1.5 +
- EJB3

3.2.2 TopCoder Software Component Dependencies:

- Base Exception 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

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
- A separate TCUML file for Cockpit Facade Util component which includes class diagrams of the existing classes (only a simple class *LoginUtil*) and new DTOs.

3.4.2 Help / User Documentation

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