

Game Plan Service 1.0 Component Specification

1. Design

The Game Plan Service component defines the interface to retrieve and aggregate data related to the game plan of TopCoder Direct Projects. The component also provides an 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 Design Patterns

Strategy pattern – this component provides GamePlanService interface and its implementation that can possibly be used in some external strategy context.

DAO pattern – GamePlanService is a DAO for TCDirectProjectGamePlanData, SoftwareProjectData and StudioProjectData DTOs.

1.2 Industry Standards

EJB3, IoC, JPA, SQL

1.3 Required Algorithms

All algorithms used in this component are quite simple and straightforward. Please see them in the implementation notes for methods in TC UML Tool.

1.3.1 Logging

This component must perform logging in all public methods of GamePlanServiceBean.

All information must be logged using log:Log attribute. If log attribute is null, then logging is not required to be performed.

In all mentioned methods method entrance with input arguments, method exit with return value and call duration time must be logged at INFO level. It's not required to log method exit when method throws an exception.

All errors (for all thrown exceptions) must be logged at ERROR level with exception message and stack trace.

1.4 Component Class Overview

GamePlanService [interface]

This interface represents a game plan service. It provides a method for retrieving game plan specific data for all TC Direct projects associated with the current user.

GamePlanServiceBean

This class is an EJB that implements GamePlanService business interface. This bean uses Logging Wrapper component to log exceptions and debug information. Also it uses injected JPA entity managers to access software and studio projects data in the persistence.

GamePlanServiceLocal [interface]

This interface represents the local interface for GamePlanService session bean. It extends that interface and provides no additional methods.

GamePlanServiceRemote [interface]

This interface represents the remote interface for GamePlanService session bean. It extends that interface and provides no additional methods.

1.5 Component Exception Definitions

GamePlanPersistenceException

This exception is thrown by GamePlanServiceBean when some error occurs while accessing the persistence.



GamePlanServiceConfigurationException

This exception is thrown by GamePlanServiceBean when error occurs while initializing this bean.

GamePlanServiceException

This exception is thrown by implementations of GamePlanService when some implementation specific error occurred. Also this exception is used as a base class for other specific custom exceptions.

1.6 Thread Safety

This component is thread safe.

Implementations of GamePlanService, GamePlanServiceLocal and GamePlanServiceRemote must be thread safe when entities passed to them are used in thread safe manner by the caller.

GamePlanServiceBean is mutable and not thread safe. But it is always used in thread safe manner in EJB container because its state doesn't change after initialization. Instances of EntityManager used by this class are thread safe.

This bean assumes that transactions are managed by the container. At the same time the bean doesn't require a transaction to be used since it's the only method doesn't modify data in the persistence.

2. Environment Requirements

2.1 Environment

Development language: Java 1.5, J2EE 1.5

Compile target: Java 1.5, J2EE 1.5

QA Environment: Solaris 7, RedHat Linux 7.1, Windows 2000, Windows 2003

2.2 TopCoder Software Components

Base Exception 2.0 – is used by custom exceptions defined in this component.

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

Cockpit Facade Util 1.1 – defines TCDirectProjectGamePlanData, SoftwareProjectData and StudioProjectData classes used in this component.

Security Manager 1.1 – defines TCSubject and RolePrincipal classes used in this component.

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

None

3. Installation and Configuration

3.1 Package Name

com.topcoder.service.gameplan

com.topcoder.service.gameplan.ejb

3.2 Configuration Parameters

3.2.1 EJB configuration of GamePlanServiceBean

GamePlanServiceBean holds 2 persistence context instances with reference names equal to "softwarePersistence" and "studioPersistence" respectively. These entity managers must be injected by EJB container.

Also GamePlanServiceBean uses the following resources injected by EJB container:

Resource name	Description	Values
logName	The name of the Logger Wrapper logger to be used by this class for logging errors and debug information. When not specified, logging is not performed.	String. Not empty. Optional.

3.3 Dependencies Configuration

Please see docs of all the dependency components to configure them properly.

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 API usage

```
// Get game plan service
Context context = new InitialContext();
GamePlanService gamePlanService =
    (GamePlanServiceRemote) context.lookup("GamePlanServiceBean/remote");

// Create TCSubject instance for admin user
Set<RolePrincipal> principals = new HashSet<RolePrincipal>();
RolePrincipal adminRole = new RolePrincipal("Cockpit Administrator", 1);
principals.add(adminRole);
TCSubject tcSubject1 = new TCSubject(principals, 12345);

// Retrieve game plan data for all existing TC Direct projects
List<TCDirectProjectGamePlanData> gamePlanDataList =
    gamePlanService.retrieveGamePlanData(tcSubject1);

for (TCDirectProjectGamePlanData gamePlanData : gamePlanDataList) {
    // Get data for software projects
    List<SoftwareProjectData> softwareProjects = gamePlanData.getSoftwareProjects();

    // Get data for studio projects
    List<StudioProjectData> studioProjects = gamePlanData.getStudioProjects();

    // Process projects data
    ...
}

// Create TCSubject instance for non-admin user
Set<RolePrincipal> principals = new HashSet<RolePrincipal>();
TCSubject tcSubject2 = new TCSubject(principals, 23456);

// Retrieve game plan data for all TC Direct projects associated with user with ID=23456
gamePlanDataList = gamePlanService.retrieveGamePlanData(tcSubject2);
```

5. Future Enhancements

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