

Direct Struts Action 3 Version 1.0 Component Specification

1. Design

As we progress in upgrading the Direct application, the next thing we are focusing on is the complete flow of launching a contest. This includes everything from entering the basic data for a contest to payment, spec review, provisioning, etc. We currently have a basic flow for launching a contest created in the service layer and being assembled for the html version of Direct. This contest will dig deeper into the process of launching a contest to address the requirements introduced and clarified by the requirements that are referenced in this contest.

This component provides the struts actions to get wiki link, get contest receipt, send contest receipt by email, view game plan, view project's contests, get active contest prize, increase prize of active contest, get active contest schedule, and extend active contest schedule.

1.1 Design Patterns

Actions can be treated like the *Controller* part of the *MVC* pattern.

Actions are also used *strategically* by the Struts framework.

1.2 Industry Standards

None.

1.3 Required Algorithms

1.3.1 Validation.

Validation here will comprise the user input as specified in the use cases provided in the ARS for each relevant action. Validation checking will be performed using Struts built-in annotations. For example such annotation would be added to the setProjectId method of the ContestReceiptRetrievalAction: @FieldExpressionValidator(key = "i18n.contestReceiptRetrievalAction.projectIdValues", fieldName = "projectId", expression = "projectId >= 0", message = "projectId must be >=0"). For the valid values see the variable docs.

1.4 Component Class Overview

ContestReceiptRetrievalAction

This action returns the contest receipt.

ActiveContestSheduleRetrievalAction

This action returns the active contest shedule.

ActiveContestSheduleExtensionAction

This action is used to extend the active contest shedule.

GamePlanRetrievalAction

This action returns the game plan.

ActiveContestPrizeRetrievalAction

This action returns the active contest prize.

ContestReceiptSendingAction

This action sends the contest receipt to the given emails.

ProjectContestRetrievalAction

This action returns the project contest and the project summary data.

WikiLinkRetrievalAction

This action returns the wiki link.

ActiveContestPrizeUpdatingAction

This action updates the active contest prize.

1.5 Component Exception Definitions

None.

1.6 Thread Safety

All the actions are not thread safe because they are mutable, but they will be used in a thread safe manner by the Struts framework. JsonResult will be used thread-safely as well.

2. Environment Requirements

2.1 Environment

- Development language: Java1.5, J2EE 1.5
- Compile target: Java1.5, J2EE 1.5
- Application Server: JBoss 4.0.2

2.2 TopCoder Software Components

- Struts Framework 1.0: defines AbstractAction, AggregateDataModel and other classes used in this design.
- Contest Service Faade 1.0: defines ContestServiceFacade and entities used in this design.
- Direct Service Faade 1.0: defines DirectServiceFacade and entities used in this design.

2.3 Third Party Components

- Struts 2.1.1 (<http://struts.apache.org/2.x/index.html>)

3. Installation and Configuration

3.1 Package Name

com.topcoder.service.actions
com.topcoder.service.actions.result

3.2 Configuration Parameters

struts.xml should be configured to map Actions to the JSP pages.

3.3 Dependencies Configuration

None.

4. Usage Notes

4.1 Required steps to test the component

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

4.2 Required steps to use the component

See demo below.

4.3 Demo

Since Struts Framework do all the work there is nothing to show in demo in fact. Here is the snippet of the JSP that will use the GamePlanRetrievalAction:

```
<%@ taglib prefix="s" uri="/struts-tags" %>
<%@ taglib prefix="sx" uri="/struts-dojo-tags" %>

<html>
<head>
<sx:head />
</head>
<body>
<s:form action="GamePlanRetrievalAction" validate="true">
<s:textfield name="projectId" label="the id of the project" />
<s:submit />
<s:reset />
</s:form>
</body>
</html>
```

After the form is submitted the game plan as List<ContestPlan> is put into ValueStack and can be obtained by the JSP. For example, if used submitted the projectId equals to 21 game plan for the project with id 21 will be returned. the The usage of other action is similar.

5. Future Enhancements

None.