

Extending CAS Using Spring Web Flow

Adam Rybicki

2010 Jasig Conference, San Diego, CA

March 8, 2010

© Copyright Unicon, Inc., 2009. This work is the intellectual property of Unicon, Inc. Permission is granted for this material to be shared for non-commercial, educational purposes, provided that this copyright statement appears on the reproduced materials and notice is given that the copying is by permission of Unicon, Inc. To disseminate otherwise or to republish requires written permission from Unicon, Inc.



1. Overview
2. What is Spring Web Flow?
3. How CAS Uses Spring Web Flow?
4. Adding to the CAS Flow
5. Adding Action Beans
6. Adding Views
7. Example Password Update Extension
8. Q/A

Overview

What Is Spring Web Flow?

How Does CAS Use Spring Web Flow?

Disclaimer

- This presentation is not a comprehensive Spring Web Flow (SWF) introduction
- It is not even a SWF overview
- The presenter is not a SWF expert
- This presentation is only meant to show how CAS uses SWF, and how you can, too.

What Is Spring Web Flow?

- Not very different in purpose from:
 - Business Process Execution Language (BPEL)
 - Various Workflow standards
- But much easier to get started with
- No huge committee behind it
 - Couple of guys, Keith Donald and Erwin Vervaet, credited for creation of SFW
- Easy to understand and become productive

What Is Spring Web Flow?

- Complements Spring Web MVC
- Allows to control the states of the application outside of its controllers
 - action state
 - decision state
 - view state
 - end state
- Upgrading from SWF 1 to 2 is supported
 - `org.springframework.webflow.upgrade.WebFlowUpgrader`

How Does CAS Use Spring Web Flow?

- Set up the initial state
- Check for the TGT
- Display the login form, if needed
- Check for the service parameter
- Redirect to service or display a generic login success message
- Set up “global” exception-handling state transitions

How Does CAS Use Spring Web Flow?

- CAS uses SWF 1
- Beginning with CAS 3.4, CAS will use Spring Framework 3 and SWF 2
- This significantly speeds up access to these new technologies over the older CAS4-based CAS roadmap
- Details in the Jasig Wiki: <http://www.jasig.org/wiki/x/VwG0AQ>

How Does CAS Use Spring Web Flow?

```
...
<action-state id="initialFlowSetup">
    <action bean="initialFlowSetupAction" />
    <transition on="success" to="ticketGrantingTicketExistsCheck" />
</action-state>

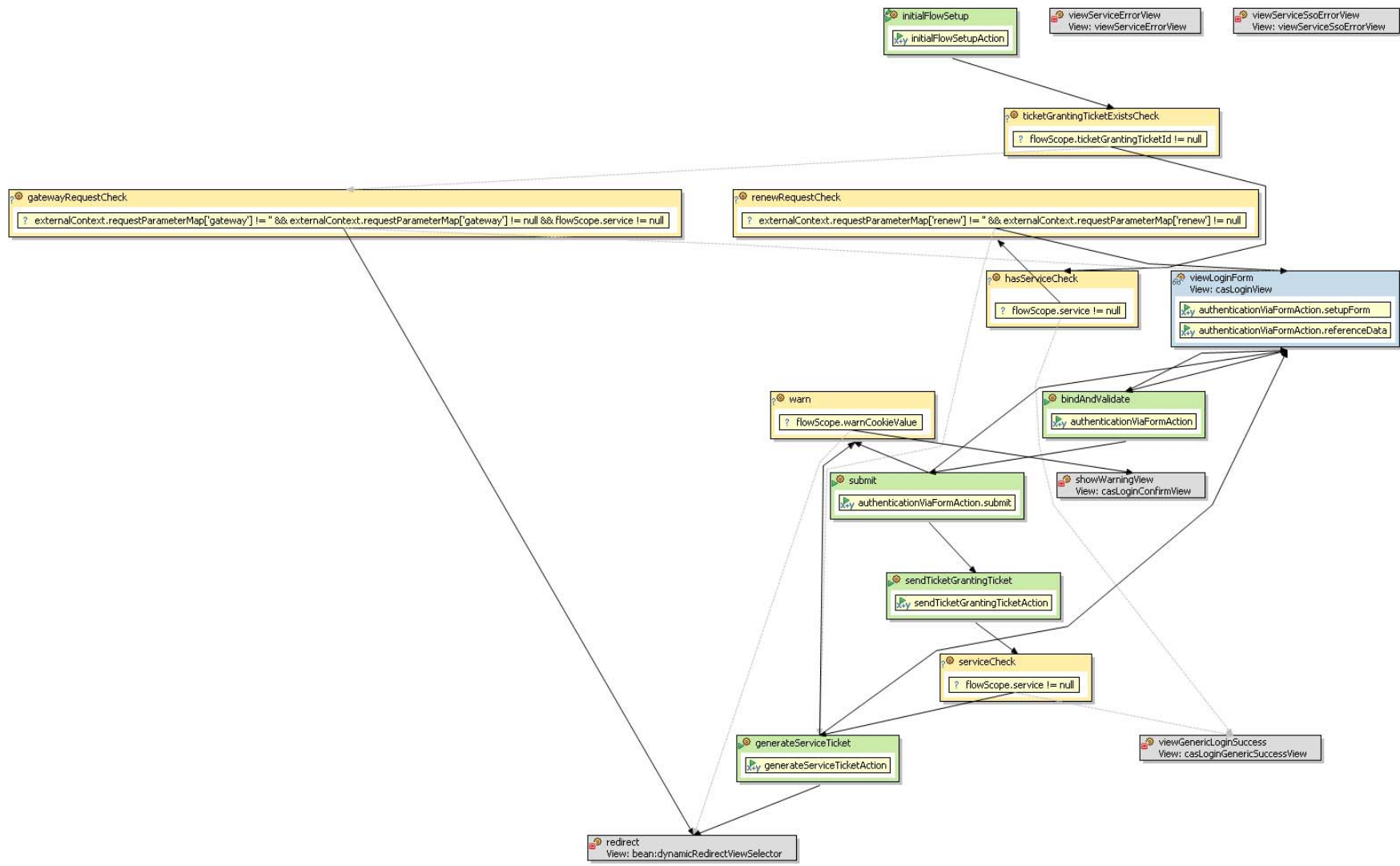
<decision-state id="ticketGrantingTicketExistsCheck">
    <if test="{flowScope.ticketGrantingTicketId != null}" then="hasServiceCheck"
        else="gatewayRequestCheck" />
</decision-state>

<decision-state id="gatewayRequestCheck">
    <if test="{externalContext.requestParameterMap['gateway'] != '' &&
        externalContext.requestParameterMap['gateway'] != null &&
        flowScope.service != null}" then="redirect" else="viewLoginForm" />
</decision-state>

<view-state id="viewLoginForm" view="casLoginView">
    <render-actions>
        <action bean="authenticationViaFormAction" method="setupForm" />
        <action bean="authenticationViaFormAction" method="referenceData" />
    </render-actions>
    <transition on="submit" to="bindAndValidate" />
</view-state>

<end-state id="redirect" view="bean:dynamicRedirectViewSelector" />
...
```

How Does CAS Use Spring Web Flow?



Adding New Functionality

Why Is This A Good Path?

Adding New Business Logic

Adding New Views

Why Use SWF to Extend CAS?

- Discrete points to inject new functionality
- The only thing to modify is the flow definition
- New beans and views can be added easily with Maven overlay build method

New Bean

- SWF-specific Spring bean type
 - Extends
`org.springframework.webflow.action.AbstractAction`
- Access to the flow scope object also accessible in the flow definition file
- `doExecute()` method contains business logic and returns `success()` or `error()` Event types
- Returned Event types are evaluated in the flow definition file

New View

- Add to `default_views.properties`
- View's name directly referenced from the flow definition file

```
### View for password update
passwordUpdateView.(class)=org.springframework.web.servlet.view.JstlView
passwordUpdateView.url=/WEB-INF/view/jsp/default/ui/passwordUpdateView.jsp
```

New Action State

- Add to `login-webflow.xml`

```
<action-state id="testPasswordUpdate">  
  <action bean="testPasswordUpdateAction" />  
  <transition on="success" to="serviceCheck" />  
  <transition on="error" to="viewPasswordUpdate" />  
</action-state>
```

Demo

Questions?



Adam Rybicki

arybicki@unicon.net

www.unicon.net



UNICON