

# Spring into Web Flow with Grails

JOSEPH FAISAL NUSAIRAT



Integrallis

Ideas, Implemented.

Ideas, Implemented.

# ABOUT ME





developing professionally since 1997  
unprofessionally since 1985

author of two Java Titles

speaker at **Jugs**, JBoss World 2008,  
a NoFluff, etc

uses mainly java, groovy, ruby

Currently playing with Objective C for the iPhone



# ABOUT THIS TALK





discuss what web flows are

what are spring web flows

show some basics of webflow

do our own live demo



# WHAT IS WEB FLOW





# what is a webflow

allows for controlling the flow of items

control is not on the page but in DSLs

can keep components open for a longer period of time





# webflows in spring

Now at version 2.0 in Spring

allows rich integration with JSF

can control scoping and conversations

cannot have multiple conversations

XML to configure





# LETS START CREATING A WEBFLOW





# creating a flow

flows live in the controller

flows will look just like an action  
except name ends with "Flow"

```
class HotelController {  
  def index = { redirect(action:'bookingCart') }  
  
  def bookingCartFlow = {  
    ...  
  }  
}
```





# defining a view

views are going to represent the individual pages

they control the actions and the possible flow points.

```
def bookingCartFlow = {  
    start() {  
    }  
}
```



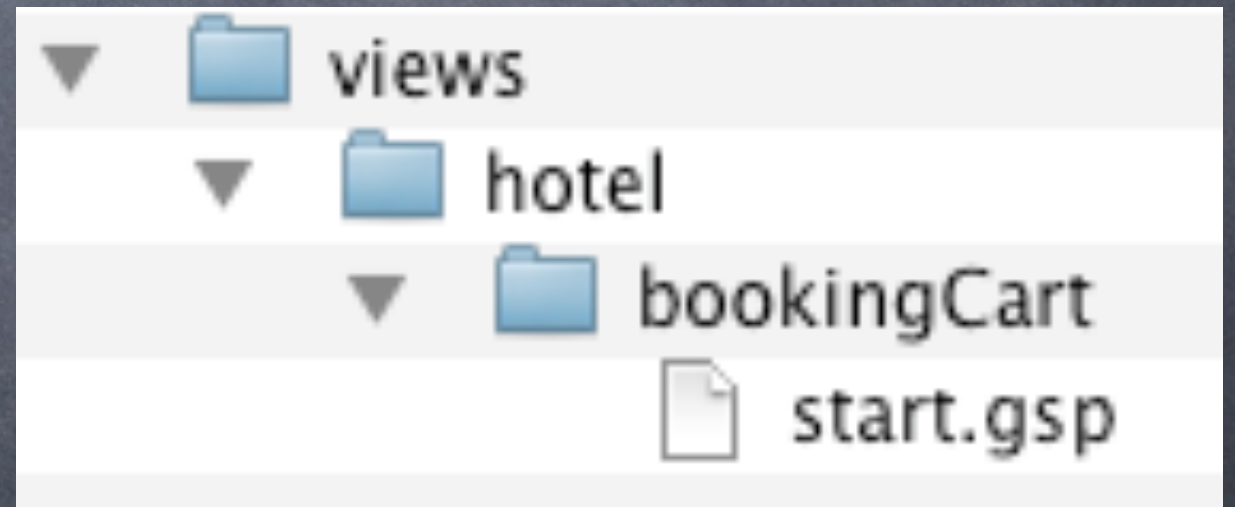


# layout for the views

each flow is a sub folder of the controller

with each view possibly being gsp page

```
class HotelController {  
  def bookingCartFlow = {  
    start() {  
    }  
  }  
}
```





# adding a default view

adding a default action

```
class HotelController {  
  def bookingCartFlow = {  
    start() {  
      action {  
        println "do some action"  
      }  
      on("success").to "findHotels"  
    }  
  }  
}
```





# adding to the view

on the page controlling the page flow

```
def bookingCartFlow = {  
  findHotels{  
    on("showHotels") {  
      def hotels = Hotel.findAllByNameIlike(  
        "%${params.hotelsName}%",  
        [max: 10])  
      [hotelResults: hotels]  
    }.to "hotelListPage"  
  }  
}
```





# dynamically altering flow

what if we want to go to a different part?

```
saveBooking() {  
  action{  
    if (!flow.booking.validate()) {  
      return bookingPage()  
    } else {  
      return confirmHotel()  
    }  
  }  
  on("confirmHotel").to "confirmHotel"  
  on("paypal").to "paypal"  
}
```





# error handling

you can catch exceptions and errors at each point too

```
class HotelController {  
  def bookingCartFlow = {  
    start() {  
      action {  
        println "do some action"  
      }  
      on("success").to "findHotels"  
      on(Exception).to "errorHandler"  
    }  
  }  
}
```





# scopes available

flash – for immediate items

flow – for the length of the flow

conversation – for the length of the flow and subflows





# CREATING A SUBFLOW





# what are subflows

ability to create alternative flow paths

call sub flows via flows

are there own flows themselves

call back to the main flow





# TESTING THE WEBFLOW





# testing the flow

testing is fully supported in Grails

test case needs to extend `grails.test.WebFlowTestCase`

overwrite the `getFlow()` method to point to our flow

submit events to the flow





```
class ShoppingCartFlowTests extends grails.test.WebFlowTestCase {  
    def getFlow() { new BookController().shoppingCartFlow }  
  
    void testShoppingCartFlow() {  
        def viewSelection = startFlow()  
        //test that you get the correct view to start assertEquals "showCart",  
        viewSelection.viewName  
  
        //set some test parameters  
        flow.params.address = new Address(...)  
  
        //signal a submission from the show cart page  
        viewSelection = signalEvent("checkout")  
        assertEquals "enterDetailsView", viewSelection.viewName  
  
        //check anything put in flash.message def message =  
        getFlowExecution().getFlashScope().get("message") assertTrue  
        message.contains("saved.")  
  
        //test your data to see that stuff worked  
        assertNotNullModelAttribute('cart',viewSelection)  
        def cart = getFlowAttribute('cart') //another way to get stuff  
        def person = viewSelection.model.cart }  
}
```





# QUESTIONS?





[WWW.INTEGRALLIS.COM](http://www.integrallis.com)

[HTTP://GITHUB.COM/INTEGRALLIS](http://github.com/integrallis)

[http://www.pragprog.com/screencasts/v-scgithub/  
insider-guide-to-github](http://www.pragprog.com/screencasts/v-scgithub/insider-guide-to-github)

