Spring Introduce Extension

John Chilton

February 27, 2008

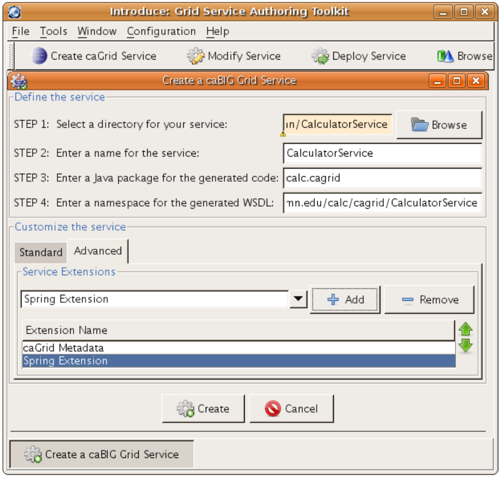
The Spring extension builds on top of the Interfaces extension providing all of the same benifits for the service developer and the user of the service.

The Spring extension provides two additional features for the service developer.

- Spring beans may be annotated as implementing interfaces for a service from inside of a Spring configuration file.
- Spring beans may be configured to act as proxies for getting and setting WSRF resource properties.

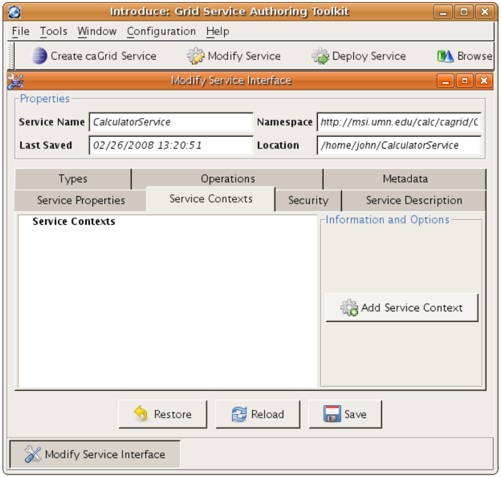
To demonstrate the Spring extension we will recreate the `CalculatorService` service, but this time we will place the operations in a service context. This will make it easier to demonstrate working with resource properties. Both extensions can be used however to manage operations on a main service and/or its service contexts.

Recreate the CalculatorService, this time with the SpringExtension.


Introduce: Grid Service Authoring Toolkit
File Tools Window Configuration Help
Create caGrid Service
Modify Service
Deploy Service
Browse
Create a caBIG Grid Service
Define the service

STEP 1: Select a directory for your service: ın/CalculatorService
STEP 2: Enter a name for the service: CalculatorService
STEP 3: Enter a Java package for the generated code: calc.cagrid
STEP 4: Enter a namespace for the generated WSDL: ꞃn.edu/calc/cagrid/CalculatorService

Extension Name
caGrid Metadata
Spring Extension

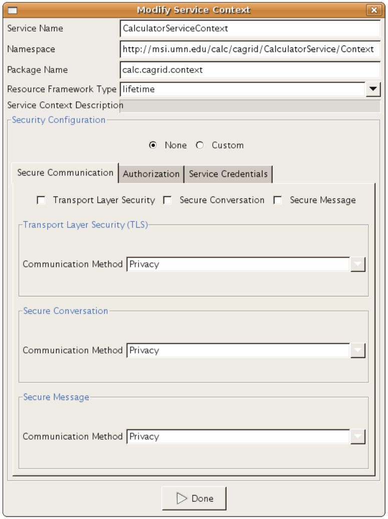


Service Name: CalculatorService
Namespace: http://msi.umn.edu/calc/cagrid/C
Last Saved: 02/26/2008 13:20:51
Location: /home/john/CalculatorService

Service Contexts

Select the "Service Contexts" tab, then click the "Add Service Context" button.

Click the “Done” button.


Modify Service Context

Service Name	CalculatorServiceContext
Namespace	http://msi.umn.edu/calc/cagrid/CalculatorService/Context
Package Name	calc.cagrid.context
Resource Framework Type	lifetime
Service Context Description	

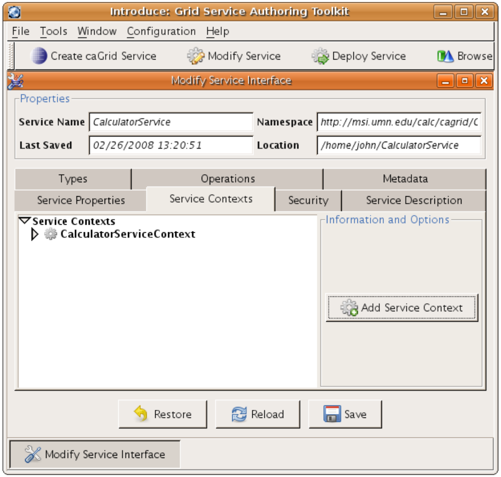
Security Configuration

[✓ None] Custom

Transport Layer Security (TLS)
Communication Method: Privacy

Secure Conversation
Communication Method: Privacy

Secure Message
Communication Method: Privacy



Service Name: CalculatorService
Namespace: http://msi.umn.edu/calc/cagrid/C
Last Saved: 02/26/2008 13:20:51
Location: /home/john/CalculatorService

Service Contexts
CalculatorServiceContext

The service is now setup.

applicationContext.xml (1 / 3)

The Spring extension creates the outline of a Spring configuration file and places it in the etc directory.

etc/applicationContext.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/...."
  xmlns:xsi="http://www.w3.org/..."
  xmlns:spring="http://msi.umn.edu/..."
  xmlns:interfaces="http://msi.umn.edu/..."
  xsi:schemaLocation="...">

  <!-- Define beans and caGrid mappings here -->
</beans>
```

applicationContext.xml (2 / 3)

Add the following lines to this file to create a calculator bean and configure it to implement Adder and Subtracter for the CalculatorServiceContext service.

etc/applicationContext.xml

```
<beans ...>
  <bean id="theCalculator" class="calc.Calculator" />
  <spring:implements-for-service
          service="CalculatorServiceContext"
          implementer="theCalculator">
    <interfaces:interface name="calc.Adder" />
    <interfaces:interface name="calc.Subtracter" />
  </spring:implements-for-service>
</beans>
```

applicationContext.xml (3 / 3)

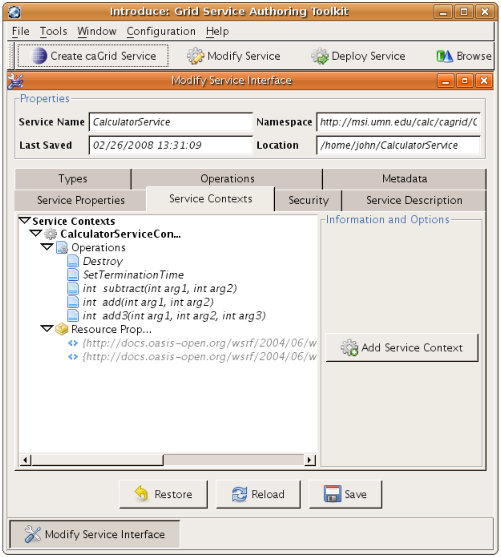
Compile Time Versus Runtime

`implements-for-service` elements are translated into service operations when Introduce saves. If any part of an `implements-for-service` element is modified the service should be resaved. If only bean definitions are modified there is no need to resave.

Importing Other Spring Files

The implementer bean definition may come from an imported Spring file, but all of the actual `implements-for-service` elements must be defined directly in the `applicationContext.xml` file.

After copying the appropriate jar and saving

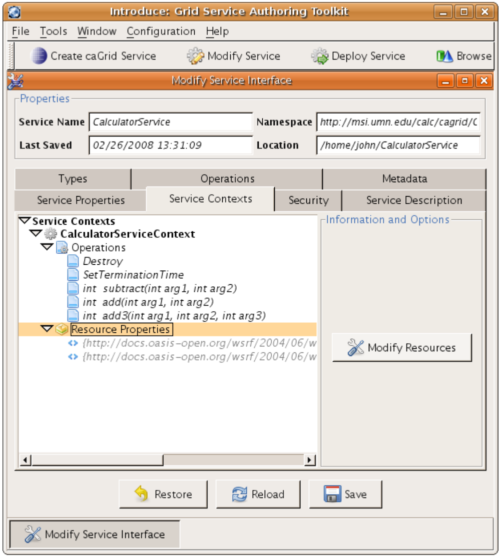


Service Name: CalculatorService
Namespace: http://msi.umn.edu/calc/cagrid/C
Last Saved: 02/26/2008 13:31:09
Location: /home/john/CalculatorService

```
Service Contexts
  CalculatorServiceCon...
    Operations
      Destroy
      SetTerminationTime
      int subtract(int arg1, int arg2)
      int add(int arg1, int arg2)
      int add3(int arg1, int arg2, int arg3)
    Resource Prop...
      {http://docs.oasis-open.org/wsrf/2004/06/w
      {http://docs.oasis-open.org/wsrf/2004/06/w
```

To demonstrate the Spring integration with resource properties we will add a resource property to the service context. We will create a string resource property that will keep track of the last operation called.

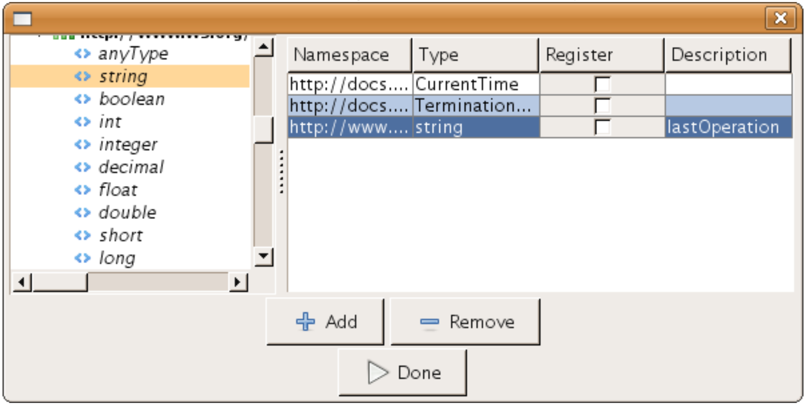
Select "Resource Properties", then click the "Modifiy Resources" button.



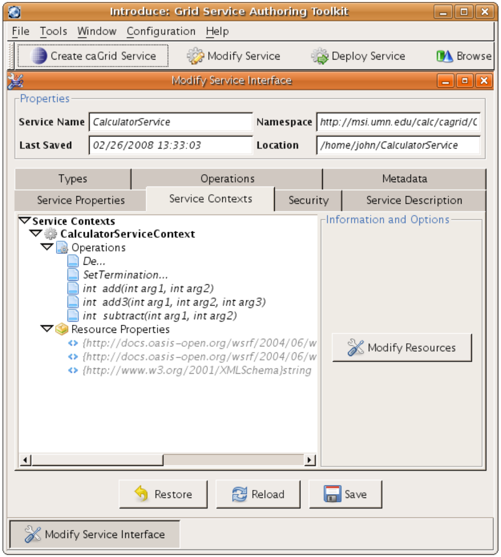
Service Name	CalculatorService	Namespace	http://msi.umn.edu/calc/cagrid/C
Last Saved	02/26/2008 13:31:09	Location	/home/john/CalculatorService

```
Service Contexts
  CalculatorServiceContext
    Operations
      Destroy
      SetTerminationTime
      int subtract(int arg1, int arg2)
      int add(int arg1, int arg2)
      int add3(int arg1, int arg2, int arg3)
    Resource Properties
      {http://docs.oasis-open.org/wsrf/2004/06/w
      {http://docs.oasis-open.org/wsrf/2004/06/w
```

Add a string resource property.



Namespace	Type	Register	Description
http://docs....	CurrentTime	☐	
http://docs....	Termination...	☐	
http://www....	string	☐	lastOperation



Service Name: CalculatorService
Namespace: http://msi.umn.edu/calc/cagrid/C
Last Saved: 02/26/2008 13:33:03
Location: /home/john/CalculatorService

```
Service Contexts
  CalculatorServiceContext
    Operations
      De...
      SetTermination...
      int add(int arg1, int arg2)
      int add3(int arg1, int arg2, int arg3)
      int subtract(int arg1, int arg2)
    Resource Properties
      {http://docs.oasis-open.org/wsrf/2004/06/w
      {http://docs.oasis-open.org/wsrf/2004/06/w
      {http://www.w3.org/2001/XMLSchema}string
```

After saving, you should see the new resource property.

Calculator.java

```
package calc;
public class Calculator implements Adder,Subtracter{
  private Tracker tracker;

  public void setTracker(Tracker tracker) {
    this.tracker = tracker;
  }

  public int add(int x, int y) {
    tracker.setLastOp("Addition");
    return x + y;
  }
}
```

Tracker.java

```
package calc;
public class Tracker {
  public String theLastOperation;
  public void setLastOp(String theLastOperation) {
    this.theLastOperation = theLastOperation;
  }
  public String getLastOp() {
    return theLastOperation;
  }
}
```

etc/applicationContext.xml

```
<bean id="theCalculator" class="calc.Calculator" >
  <property name="tracker">
    <bean class="calc.Tracker">
      <spring:resource-property name="string"
        get-method="getLastOp"
        set-method="setLastOp" />
    </bean>
  </property>
</bean>
```

The `resource-property` elements says to override this beans `getLastOp` and `setLastOp` methods to get and set the resource property `string`.

It should be noted that the Tracker class field theLastOperation does not become the resource property. The Tracker class is subclassed at runtime and setLastOp and getLastOp are overridden to access the resource property.

As illustratation, this Tracker definition would also work.

Tracker.java

```
package calc;
abstract public class Tracker {
  abstract public void setLastOp(String lastOp);
  abstract public String getLastOp();
}
```

resource-property and Threads

Like session or request scoped Spring beans, the behavior overridden methods will be tied to the thread they are invoked from. The thread determines what endpoint reference was invoked and hence what resource to access.

Warning

If these methods are invoked from a thread that doesn't corrspepond to an Axis request or to a service without the specified resource property, a runtime exception will be thrown.

resource-property

Any number of beans can expose any number of resource properties, and any resource property can be exposed via any number of beans. The `get-method` or `set-method` attributes may be excluded, but not both.