|  |  |  |
| --- | --- | --- |
| doc-header-print-portrait | |  |
|  | | Engineering Design Document |
|  | | Validation Logic Integration into Activity Services |
|  | **Version: 1.0**  **Last Revised:** March 16, 2011  **Author: Borys Sokolov** | |

Table of Contents

[1 Problem Overview 4](#_Toc287975649)

[2 Proposed Solution 4](#_Toc287975650)

[3 Functional Design 4](#_Toc287975651)

[4 Technical Design 4](#_Toc287975652)

[4.1 Inspector 4](#_Toc287975653)

[4.2 Attribute to inject inspector 4](#_Toc287975654)

[4.3 Sample code 4](#_Toc287975655)

[5 Error list 6](#_Toc287975656)

[6 Outstanding Issues 6](#_Toc287975657)

Document Version History

|  |  |  |  |
| --- | --- | --- | --- |
| **Author** | **Revision No.** | **Date** | **Description of Change** |
| **Borys Sokolov** | 1.0 | 03/15/2011 | Initial version |
|  |  |  |  |
|  |  |  |  |

Functional Design Review

| **Name** | **Title** | **Date Reviewed** | **Date Approved** |
| --- | --- | --- | --- |
| **Kyle Quest** | Architect |  |  |
| **Anatoliy Lokshin** | Development lead | 03/16/2011 | 03/16/2011 |
| **Yulia Kuchmai** | QA Representative |  |  |
|  |  |  |  |

Technical Design Review

| **Name** | **Title** | **Date Reviewed** | **Date Approved** |
| --- | --- | --- | --- |
| **Kyle Quest** | Architect |  |  |
|  |  |  |  |
|  |  |  |  |

Engineering Task Worklist Review

| **Name** | **Title** | **Date Reviewed** | **Date Approved** |
| --- | --- | --- | --- |
| **Anatoliy Lokshin** | Development lead |  |  |
|  |  |  |  |
|  |  |  |  |

References

|  |  |  |
| --- | --- | --- |
| **Document Name** | **Author** | **Location** |
| MetraTech Security Framework Specification | Kyle Quest | <svn://qaautoserv/SecurityFramework/branches/SF1.0-Development/Docs/MtSecurityFramework.doc> |
|  |  |  |

Version Configuration

|  |  |  |
| --- | --- | --- |
| **Name** | **Version** | **Additional Comments** |
|  |  |  |
|  |  |  |

# Problem Overview

Validation logic should be integrated to the Activity Services. It is preferable to use no direct cutting into the method code.

# Proposed Solution

We propose to create custom Inspector Attribute which could be added to the specific methods to extend them with validation functionality. In this case we could locate all the validation functionality in that inspector and not touch the method’s sources (only adding specified attribute).

# Functional Design

The main validation functionality should be placed in the Inspector. All needed info for parsing method params should be sent there.

There should be the possibility to set which params should be validated. By default Inspector should run validation for all the params.

Inspector should have a posiblity to check composite types.

# Technical Design

## Inspector

Inspector should implement System.ServiceModel.Dispatcher.IParameterInspector interface. It would afford and opportunity to cut into before and after the call of the method without any changes to the method’s source. There are only two methods to implement. For parameter validation public object BeforeCall(string operationName, AfterCall [] inputs) should be used. Only name specified by the OperationContract attribute (but not full method signature) would pass to the operationName. Array inputs contains the values to inspect. BeforeCall returns an object value. By this value it is only possible to correlate BeforeCall and AfterCall methods (it would pass to the correlationState param of the AfterCall) and there is no other usage for it. So in our case this value wouldn’t influence on the service work and could have any value or even be a null.

All the exceptions that could rise during the validation process should be wrapped as MASBasicException for correct error handling on the client side.

## Attribute to inject inspector

Developed Inspector could be injected into the Service as an Attribute. This Attribute should implement System.ServiceModel.Description.IOperationBehavior interface. In public void ApplyDispatchBehavior(OperationDescription operationDescription, DispatchOperation dispatchOperation) current Inspector could be added. Only this method is useful here. operationDescription contains full operation information and could be sent to the Inspector.

## Implementation in Service

Attribute should be added to the methods which require validation.

## Sample code

### Inspector

public class MyParameterInspector : IParameterInspector

{

public void AfterCall(string operationName, object[] outputs, object returnValue, object correlationState)

{

}

public object BeforeCall(string operationName, object[] inputs)

{

//check routine

return null;

}

public System.ServiceModel.Description.OperationDescription OperationDescription

{ get; set; }

}

### attribute

public class MyInspectorAttribute : Attribute, IOperationBehavior

{

public void AddBindingParameters(OperationDescription operationDescription, System.ServiceModel.Channels.BindingParameterCollection bindingParameters)

{

}

public void ApplyClientBehavior(OperationDescription operationDescription, System.ServiceModel.Dispatcher.ClientOperation clientOperation)

{

}

public void ApplyDispatchBehavior(OperationDescription operationDescription, System.ServiceModel.Dispatcher.DispatchOperation dispatchOperation)

{

MyParameterInspector inspector = new MyParameterInspector();

inspector.OperationDescription = operationDescription;

dispatchOperation.ParameterInspectors.Add(inspector);

}

public void Validate(OperationDescription operationDescription)

{

}

}

### Service

[ServiceContract]

public interface IService1

{

[OperationContract]

[MyInspectorAttribute]

string GetData(int value);

[OperationContract]

CompositeType GetDataUsingDataContract(CompositeType composite);

}

# Error list

The following table shows a listing of errors that can occur.

|  |  |  |  |
| --- | --- | --- | --- |
| **Error Code** | **Error Message** | **Description** | **Area** |
| MASBasicException | Invalid parameter *ParamName* | Validation or one of the input parameters failed |  |
|  |  |  |  |

# Outstanding Issues

List all open issues regarding this document.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | **Date raised** | **Description and Resolution** | **Page/ Section** | **Raised by** | **Allocated to** | **Status** |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |