



WSO2 API Manager 4.2.0 Advanced - Integration Profile

Extending the Micro Integrator

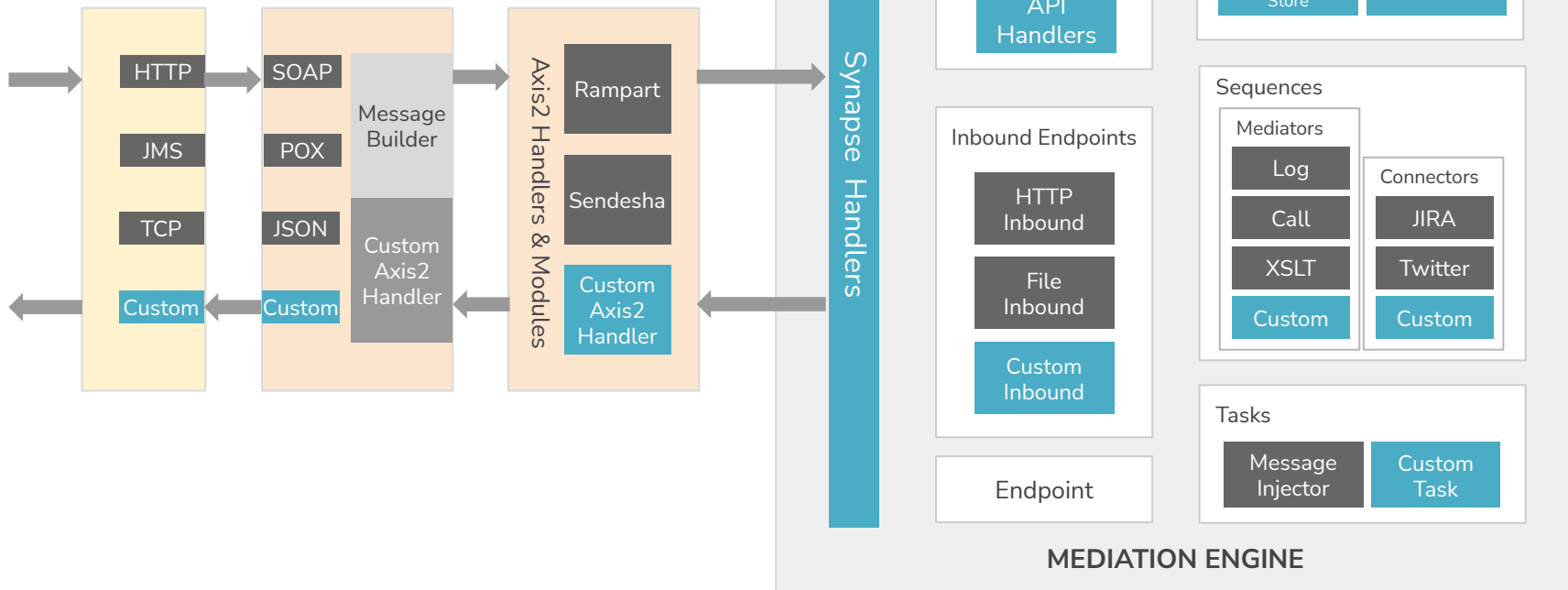


WSO2 Training

Module Outline

- Extension points in the Micro Integrator
- Script Mediator
- Custom Mediator
- Custom Task
- Custom Connector
- Custom Message Builders and Formatters

Extension Points



Extension Points

- Script mediator
- Custom/Class mediators
- API handlers
- Custom tasks
- Custom Connectors
- Custom inbound endpoints
- Custom message stores/processors
- Custom XPath functions
- Synapse handlers
- Axis2 handlers
- Axis2 modules
- Custom message builders and formatters
- Transport listeners and senders



Script Mediator

- Use scripting languages to write mediation logic.
- Supports all languages compatible with the Bean Scripting Framework (BSF):
 - ⦿ JavaScript
 - ⦿ Ruby
 - ⦿ Groovy
- Some languages require deploying third-party libraries.

Script Mediators

Custom Mediator

- Fundamental construct of WSO2 Micro Integrator.
- Implements a standard Java interface.
- Implements the mediator interface or extends from AbstractMediator.
- Changes the message context or performs internal/external operations within the message flow.
- Returns false or throws a SynapseException to end the message flow

```
<class name="com.wso2.mediators.Test">  
    <property name="foo" value="bar"/>  
</class>
```

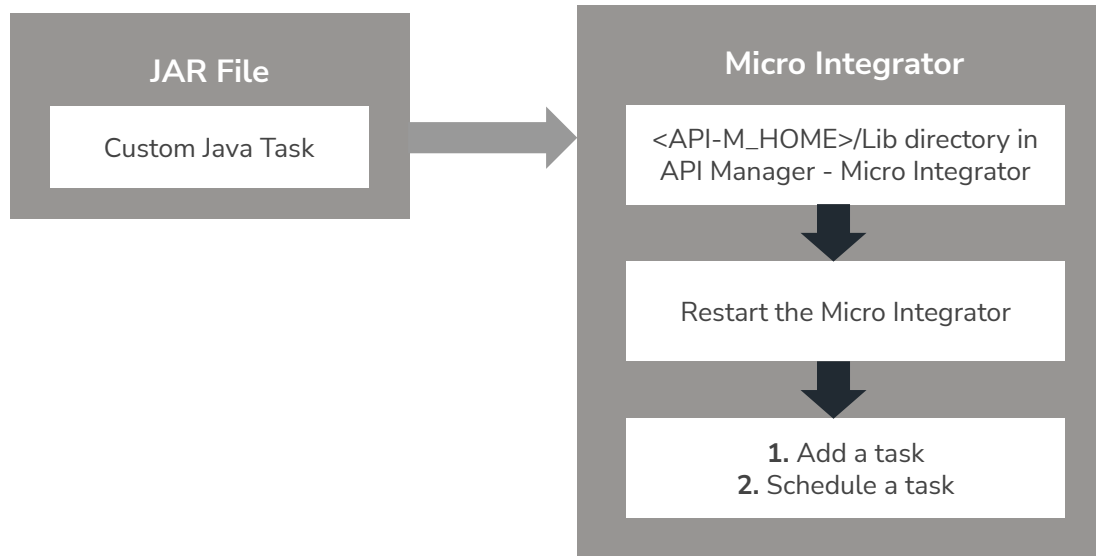
Let's try it out!

Creating a Custom Mediator



Custom Task

Write your own task-handling implementation for specific requirements.



Let's try it out!

Creating a Custom Task



Custom Connectors

- Connectors allow your message flows to interact with services such as Twitter, Salesforce, Google Docs, and JIRA.



- You can also create your own connectors and make them available to other users as ZIP files.

Let's try it out!

Creating a Custom Connector



Custom Message Builders/Formatters

Create custom message builders and formatters to manipulate incoming and outgoing messages.

1. Build a JAR file with your message builder/formatter implementation and add to the `<ML_HOME>/lib` directory.
2. Update the product configuration file (*deployment.toml* file).

Example

```
[[custom_message_builders]]
content_type = "text/xml"
class="org.test.builder.http.CustomBuilderForTextXml"

[[custom_message_formatters]]
content_type = "text/xml"
class="org.apache.axis2.transport.http.SOAPMessageFormatter"
```