

Embedding

Embedding

ServiceMix can be easily embedded in any kind of application (web or not). The most common way to do that is to use a single file to configure the ServiceMix container, its components and endpoints. Of course you can also configure ServiceMix programmatically, which has proven to be useful in some scenarios (and for unit testing).

The 3.1.1 version requires the following libraries at a bare minimum (you will have to add any libraries required by the components you use of course):

- servicemix-core
- servicemix-jbi
- servicemix-services
- backport-util-concurrent
- spring
- xbean-spring

The distribution contains several examples of such configurations files (refer to the [Basic](#) example).

Easier configuration

Starting from ServiceMix 3.2, an easier configuration of components and endpoints can be used.

Usually, all the configuration parameters are specified on the endpoints and nothing on the components. Thus the components and endpoints have been separated, and components are now optional.

The following configuration defines a JSR181 and an HTTP endpoint and will automatically create the needed components to host them:

Components omitted

```
<beans xmlns:sm="http://servicemix.apache.org/config/1.0"
      xmlns:jsr181="http://servicemix.apache.org/jsr181/1.0"
      xmlns:http="http://servicemix.apache.org/http/1.0"
      xmlns:test="urn:test">
  <sm:container id="jbi" embedded="true">
    <sm:endpoints>
      <jsr181:endpoint
        pojoClass="org.apache.servicemix.itests.beans.Echo"
        service="test:Hello"
        endpoint="testService" />
      <http:endpoint
        service="test:Hello"
        endpoint="testService"
        role="consumer"
        defaultOperation="echo"
        locationURI="http://localhost:8194/Service/"
        soap="true" />
    </sm:endpoints>
  </sm:container>
</beans>
```

If you need to configure a component specifically, you can do it as shown in the following snippet:

Components omitted

```
<beans xmlns:sm="http://servicemix.apache.org/config/1.0"
       xmlns:jsr181="http://servicemix.apache.org/jsr181/1.0"
       xmlns:http="http://servicemix.apache.org/http/1.0"
       xmlns:test="urn:test">
  <sm:container id="jbi" embedded="true">
    <sm:components>
      <http:component>
        <http:configuration>
          ...
        </http:configuration>
      </http:component>
    </sm:components>
    <sm:endpoints>
      <http:endpoint
        service="test:Hello"
        endpoint="testService"
        role="consumer"
        defaultOperation="echo"
        locationURI="http://localhost:8194/Service/"
        soap="true" />
    </sm:endpoints>
  </sm:container>
```

ServiceMix will automatically choose the configured component for all supported endpoints. If you want to define several components with different configurations and choose which component an endpoint should be deployed onto, you need to explicitly name the components and link the endpoints:

Several components

```
<beans xmlns:sm="http://servicemix.apache.org/config/1.0"
       xmlns:jsr181="http://servicemix.apache.org/jsr181/1.0"
       xmlns:http="http://servicemix.apache.org/http/1.0"
       xmlns:test="urn:test">
  <sm:container id="jbi" embedded="true">
    <sm:components>
      <http:component name="http1">
        ...
      </http:component>
      <http:component name="http2">
        ...
      </http:component>
    </sm:components>
    <sm:endpoints>
      <http:endpoint component="http1" ... />
      <http:endpoint component="http2" ... />
    </sm:endpoints>
  </sm:container>
```