

ASSESSMENT - 06

Agateeswaran K

ACE11689

Task: XSLT Mediator

API Name: XSLT_API

Description:

1) Create a REST API to accept XML request, perform XSLT functions (Concat, substring-after, count, replace) and respond to client.

Task-01

Steps:

1) To call the API using POST Method.

2) Read the request payload using the property mediator.

3) After reading the data successfully, then use the XSLT Mediator and perform the operations (Concat, substring-after, count, replace, Upper) and print the value in response.

Sample Payload Request:

```
<root>
```

```
  <Employee>
```

```
    <FirstName>John</FirstName>
```

```
    <LastName>Michael</LastName>
```

```
    <Role>Manager</Role>
```

```
    <EmployeeId>1133</EmployeeId>
```

```
    <Salary>25067.13</Salary>
```

```
  </Employee>
```

```
</root>
```

Code:

XSLT_API.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<api context="/xslt" name="XSLT_API" xmlns="http://ws.apache.org/ns/synapse">
  <resource methods="POST">
    <inSequence>
      <property expression="$body/*" name="inputPayload" scope="default"
type="STRING"/>
      <xslt key="PayloadTransformer_LE"/>
      <respond/>
    </inSequence>
    <outSequence/>
    <faultSequence/>
  </resource>
</api>
```

PayloadTransformer_LE.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<localEntry key="PayloadTransformer_LE" xmlns="http://ws.apache.org/ns/synapse">
  <xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:template match="/">
      <xsl:variable name="fname" select="//FirstName"/>
      <xsl:variable name="lname" select="//LastName"/>
      <xsl:variable name="role" select="//Role"/>
      <xsl:variable name="empId" select="//EmployeeId"/>
      <xsl:variable name="salary" select="//Salary"/>
      <output>
        <transformedData>
          <concatenatedName>
            <xsl:value-of select="concat($fname, ' ', $lname)"/>
          </concatenatedName>
          <roleAfterSubstring>
            <xsl:value-of select="substring-after($role, 'n')"/>
          </roleAfterSubstring>
          <employeeIdCount>
            <xsl:value-of select="count($empId)"/>
          </employeeIdCount>
          <replacedName>
            <xsl:value-of select="replace($fname, 'John', 'Jane')"/>
          </replacedName>
          <RoleUpper>
            <xsl:value-of select="translate($role, 'abcdefghijklmnopqrstuvwxyz',
'ABCDEFGHIJKLMNOPQRSTUVWXYZ')"/>
          </RoleUpper>
        </transformedData>
      </output>
    </template>
  </xsl:stylesheet>
</localEntry>
```

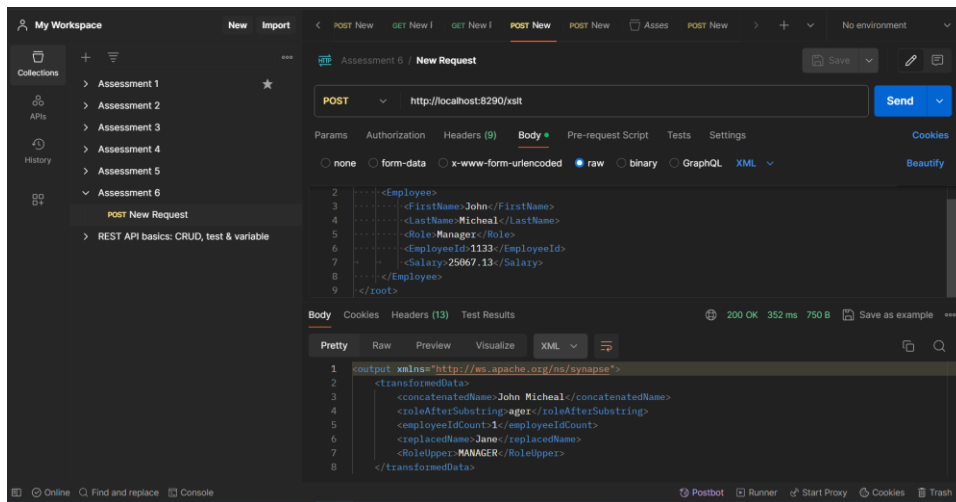
```

    </RoleUpper>
  </transformedData>
</output>
</xsl:template>
</xsl:stylesheet>
</localEntry>

```

Output:

Success Case:



Error Case:

