**Value: Logout**

### **Problem:**

User able to do the logout after successful login.

### **Templates:**

Travelocity web App logout

Playground web App logout

PassiveSTS web App logout

### **Steps:**

Step 01: User create

Step 02: SP caretion

Step 03: User login to web application

Step 04: User logout from web application

**Preconditions:**

* User needs to deploy travelocity, playground and passiveSTS webapps in tomcat server.Check [1] on how to build webapps from repository.
* Wso2 IS server should up and running

[1]<https://docs.wso2.com/display/IS541/Downloading+a+Sample>

### **Configurations**

1. Go to user.properties file and change the values of configurations as below.

#### **IS Server**

serverHost=< HOST NAME OF IS SERVER >

serverPort=< PORT OF THE IS SERVER >

scenarioName=< Scenario Name >

#### **User Management**

adminusername=< ADMIN USER NAME >

adminpassword=< ADMIN PASSWORD >

adminCredentials=< BASE64 Encoded ADMIN USERNAME:ADMIN PASSWORD >

rolename= < NEW ADMIN ROLE NAME >

usernamePrefix=< NEW ADMIN USERNAME >

#### **SP**

spname=< SP NAME >

spdescription=< SP DESCRIPTION >

carbonServer=< ISSUER NAME >

## **Travelocity**

travelocityAppName=travelocity.com spName=< SP NAME OF TRAVELOCITY >

spDescription=< SP DESCRIPTION OF TRAVELOCITY >

## **Playground**

playgroundHost=< HOST NAME FOR PLAYGROUND APP >

spName2=< SP NAME OF PLAYGROUND >

spDescription2=< SP DESCRIPTION OF PLAYGROUND >

## **PssiveSTS**

spName3=< SP NAME OF passiveSTS >

spDescription3=< SP DESCRIPTION OF passiveSTS >

**Value: Multi Factor Authentication -Linkedin connector**

### **Problem:**

By default basic authentication (with username and password) available in the system login. To increase the security of the system, it’s configured with the extra authentication step. Here

Linkedin authenticator configured as the federated authenticator in the wso2 IS and Linkedin users can login to the system by their Linkedin user credentials.

### **Templates:**

Travelocity application with MFA

Playground application with MFA

**Steps:**

Step 02: SP creation

Step 03: User login to web application

Authentication step 01: Basic authentication

Authentication step 02: Linkedin federate authentication

Steo 03: Delete SP and IDP

**Preconditions:**

* User needs to deploy travelocity.
* Wso2 IS server should up and running
* Likedin authenticator jar file should place in the <IS\_HOME>/repository/components/dropins

### **Configurations**

1. Go to user.properties file and change the values of configurations as below.

#### **IS Server**

serverHost=< HOST NAME OF IS SERVER >

serverPort=< PORT OF THE IS SERVER >

scenarioName=< Scenario Name >

#### **User Management**

adminusername=< ADMIN USER NAME >

adminpassword=< ADMIN PASSWORD >

adminCredentials=< BASE64 Encoded ADMIN USERNAME:ADMIN PASSWORD >

## **Travelocity**

travelocityAppName=travelocity.com spName=< SP NAME OF TRAVELOCITY >

spDescription=< SP DESCRIPTION OF TRAVELOCIT>

spName=< SP NAME OF TRAVELOCITY >

spDescription=< SP DESCRIPTION OF TRAVELOCITY >

## **Playground**

playgroundHost=< HOST NAME FOR PLAYGROUND APP >

spName2=< SP NAME OF PLAYGROUND >

spDescription2=< SP DESCRIPTION OF PLAYGROUND >

**Connectors (Linkedin)**

LinkedinUserName=<LINKEDIN USER NAME>

LinkedinPassword=<LINKEDIN PASSWORD>

linkdinCallBackUrl =<LINKEDIN CALLBACK URL>

linkedinClientIdl= <LINKEDIN CLIENT ID>

linkedinClientSecret=<LINKEDIN CLIENT SECRET>

**Value: Attribute profile enabled SP**

### **Problem:**

The Identity Server supports a basic attribute profile where the identity provider can include the user’s attributes in the SAML Assertions as an attribute statement. You can define the claims that must be included under service provider claim configurations. Also, once you select the “Include Attributes in the Response Always” checkbox, the identity provider always includes the attribute values related to selected claims in the SAML Attribute statement.

### **Templates:**

Travelocity web app login

### **Steps:**

Step 01: User creation

Step 02: SP creation with enable attribute profile and define claim

Step 03: Travelocity web app login

Step 04: Delete users and SP

**Preconditions:**

* User needs to deploy travelocity
* Wso2 IS server should up and running

### **Configurations**

#### **IS Server**

serverHost=< HOST NAME OF IS SERVER >

serverPort=< PORT OF THE IS SERVER >

scenarioName=< Scenario Name >

#### **User Management**

adminusername=< ADMIN USER NAME >

adminpassword=< ADMIN PASSWORD >

adminCredentials=< BASE64 Encoded ADMIN USERNAME:ADMIN PASSWORD >

rolename= < NEW ADMIN ROLE NAME >

usernamePrefix=< NEW ADMIN USERNAME >

userEmailAddress=<USER EMAIL ADDRESS>

#### **SP**

spname=< SP NAME >

spdescription=< SP DESCRIPTION >

carbonServer=< ISSUER NAME >

## **Travelocity**

travelocityAppName=travelocity.com spName=< SP NAME OF TRAVELOCITY >

spDescription=< SP DESCRIPTION OF TRAVELOCITY >

**Value: SP configuration with urls**

### **Problem:**

Metadata for a service provider may be published in a well known location via a URI. This option allows you to provide the configuration data required for configuring SAML2, by providing a URI(Ex: "https://spconfigs.com/sample-sp.xml") instead of having to manually enter the values. This enables faster entry of configuration data and allows the user to use the same metadata XML file for multiple instances of entity configuration.

### **Templates:**

Travelocity SP creation

### **Steps:**

Step 01: Sp creation

Step 02: SP configuration with url

**Preconditions:**

* User needs to deploy travelocity
* Wso2 IS server should up and running
* Metadata file should place in the <tomcathome>/webapp
* Tomcat server should up

### **Configurations**

#### **IS Server**

serverHost=< HOST NAME OF IS SERVER >

serverPort=< PORT OF THE IS SERVER >

scenarioName=< Scenario Name >

#### **User Management**

adminusername=< ADMIN USER NAME >

adminpassword=< ADMIN PASSWORD >

adminCredentials=< BASE64 Encoded ADMIN USERNAME:ADMIN PASSWORD >

rolename= < NEW ADMIN ROLE NAME >

usernamePrefix=< NEW ADMIN USERNAME >

userEmailAddress=<USER EMAIL ADDRESS>

#### **SP**

spname=< SP NAME >

spdescription=< SP DESCRIPTION >

carbonServer=< ISSUER NAME >

## **Travelocity**

travelocityAppName=travelocity.com spName=< SP NAME OF TRAVELOCITY >

spDescription=< SP DESCRIPTION OF TRAVELOCITY >

#### **IS Server**

serverHost=< HOST NAME OF IS SERVER >

serverPort=< PORT OF THE IS SERVER >

scenarioName=< Scenario Name >

#### **User Management**

adminusername=< ADMIN USER NAME >

adminpassword=< ADMIN PASSWORD >

adminCredentials=< BASE64 Encoded ADMIN USERNAME:ADMIN PASSWORD >

rolename= < NEW ADMIN ROLE NAME >

usernamePrefix=< NEW ADMIN USERNAME >

userEmailAddress=<USER EMAIL ADDRESS>

#### **SP**

spname=< SP NAME >

spdescription=< SP DESCRIPTION >

carbonServer=< ISSUER NAME >

## **Travelocity**

travelocityAppName=travelocity.com spName=< SP NAME OF TRAVELOCITY >

spDescription=< SP DESCRIPTION OF TRAVELOCITY >

metadataFilePath=<METADATA FILE URL>

managmentConsole=<CARBON>

spListUrlCarbon=<SERVICE PROVIDER LIST JSP>

loadSpUrlCarbon=<LOAD SERVICE PROVIDER JSP>

loadSpUrlCarbon=<ADD SERVICE PROVIDER LIST>

**Value: Saas enable SP**

### **Problem:**

The SaaS Application configuration defines which users you want to be able to log into your web application.

Enabled the SaaS Application checkbox, means web application is shared among tenants so users from any tenant will be allowed to log into the web application.

### **Templates:**

Travelocity and playground SP creation one tenant and users create in another tenant

### **Steps:**

Step 01: User create in tenant A

Step 02: Sp carete in tenant B

Step 03: Tenant A user access the tenant B sp

Step 04: delete users and tenants

**Preconditions:**

* User needs to deploy travelocity
* Wso2 IS server should up and running
* <TOMCAT\_HOME>/webapps/travelocity.com/WEB­INF/classes/travelocity.properties file ‘QueryParams=tenantDomain=tenant.domain’ uncomment and add the tenant domain

### **Configurations**

#### **IS Server**

serverHost=< HOST NAME OF IS SERVER >

serverPort=< PORT OF THE IS SERVER >

scenarioName=< Scenario Name >

#### **User Management**

adminusername=< ADMIN USER NAME >

adminpassword=< ADMIN PASSWORD >

adminCredentials=< BASE64 Encoded ADMIN USERNAME:ADMIN PASSWORD >

rolename= < NEW ADMIN ROLE NAME >

usernamePrefix=< NEW ADMIN USERNAME >

adminusernameTa=<TENAT ‘A’ ADMIN USER NAME>

adminpsswordTa=<TENANT ‘A’ ADMIN PASSWORD>

emailAddressTa=<TENANT ‘A’ EMAIL ADDRESS>

fNameTa=<TENAT ‘A’ USER FIRST NAME>

lNameTa=<TENANT ‘A’ USER LAST NAME>

domainTa=<TENANT ‘A’ DOMAIN ID>

adminusernameTb=<TENAT ‘B’ ADMIN USER NAME>

adminpsswordTb=<TENANT ‘B’ ADMIN PASSWORD>

emailAddressTb=<TENANT ‘B’ EMAIL ADDRESS>

fNameTb = <TENAT ‘B’ USER FIRST NAME>

lNameTb = <TENANT ‘B’ USER LAST NAME>

domainTb = <TENANT ‘B’ EMAIL ADDRESS>

adminusernameLoginTa =< TENANT ‘A’ USER LOGIN NAME>

adminusernameLoginTb = < TENANT ‘B’ USER LOGIN NAME>

#### **SP**

spname=< SP NAME >

spdescription=< SP DESCRIPTION >

carbonServer=< ISSUER NAME >

## **Travelocity**

travelocityAppName=travelocity.com spName=< SP NAME OF TRAVELOCITY >

spDescription=< SP DESCRIPTION OF TRAVELOCITY >

## **Playground**

playgroundHost=< HOST NAME FOR PLAYGROUND APP >

spName2=< SP NAME OF PLAYGROUND >

spDescription2=< SP DESCRIPTION OF PLAYGROUND >