Solution 1: Configure sending a message

1. **Create a web service collection.**

   a) Right-click on **si.ta.messaging.fraudcheck** package and select **New ➔ Web Service Collection**.

   b) Enter **fraudcheckwsc** as the new WS-I web service collection name.

   c) Click the **PLUS** symbol to add a new resource.

   d) Enter the resource URL.

   http://localhost:8890/ab/ws/externalapp/webservice/FraudReportAPI?WSDL

   e) Click the **Fetch** icon to retrieve external resources. Make sure ExternalApp is running.

2. **Create message transport plugin class that implements InitializablePlugin interface.**

   a) Right-click on **si.ta.messaging.fraudcheck** package and select **New ➔ Gosu Class**.

   b) Enter **FraudCheckTransport** as the new class name.

   c) Implement the **MessageTransport** and **InitializablePlugin** interfaces.

   d) Configure **set Parameters** property.

   e) Configure the **send** method.

```
package si.ta.messaging.fraudcheck

uses gw.plugin.InitializablePlugin
uses gw.plugin.messaging.MessageTransport
uses si.ta.messaging.fraudcheck.fraudcheckwsc.fraudreportapi.FraudReportAPI

class FraudCheckTransport implements MessageTransport, InitializablePlugin{
  // Plugin parameters
  private static final var USERNAME = "Username"
  private static final var PASSWORD = "Password"
  // API parameters
  private static var _username : String
  private static var _password : String

  var _fraudReportApi: FraudReportAPI

  override function send(message : Message, transformedPayload : String) {
    var ackCode = _fraudReportApi.checkForFraudReport(transformedPayload)
  }

  override function shutdown() {  }

  override function suspend() {
    _fraudReportApi = null
  }

  override function resume() {
    _fraudReportApi = new FraudReportAPI()
    _fraudReportApi.Config.Http.Authentication.Basic.Username = _username
    _fraudReportApi.Config.Http.Authentication.Basic.Password = _password
  }

  override property set DestinationID(destinationID : int) {  }

  override property set Parameters(map : Map<Object, Object>) {
    _username = map.get(USERNAME) as String
```

```
    _password = map.get(PASSWORD) as String
  }

}
```

3. **Create and configure transport plugin registry.**

   a) Navigate to **configuration.config.Plugins.registry** package.

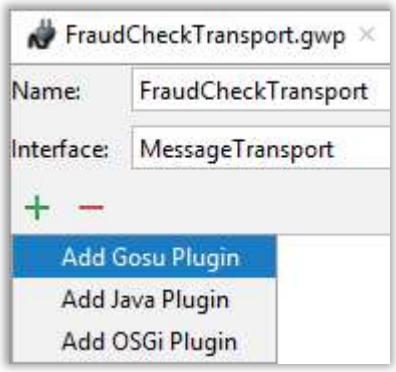
   b) Right-click on **registry** and select **New ➔ Plugin**.

   c) Enter **FraudCheckTransport** for the plugin name.

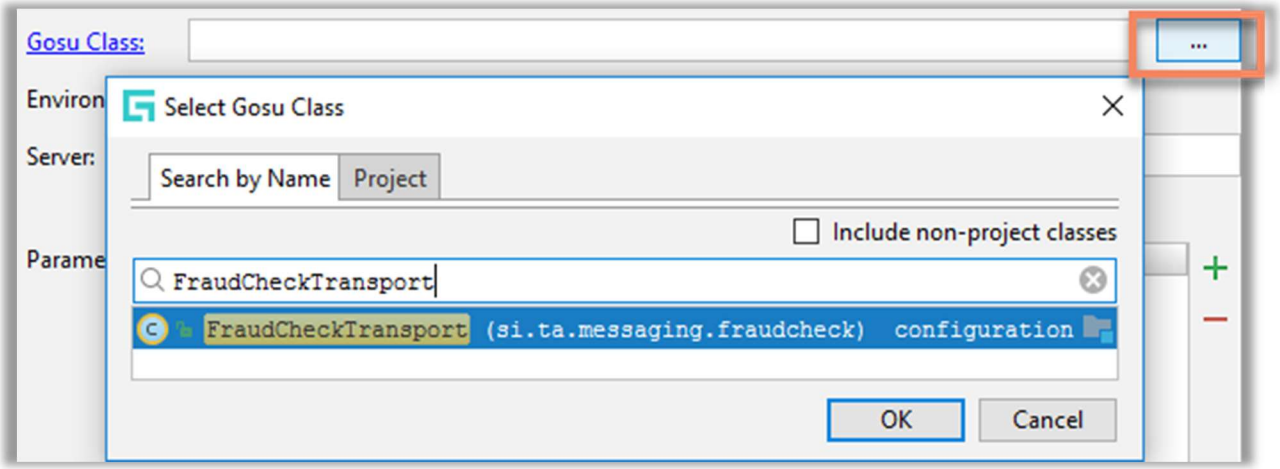
   d) Enter **MessageTransport** for the Interface.

   e) Configure the plugin.

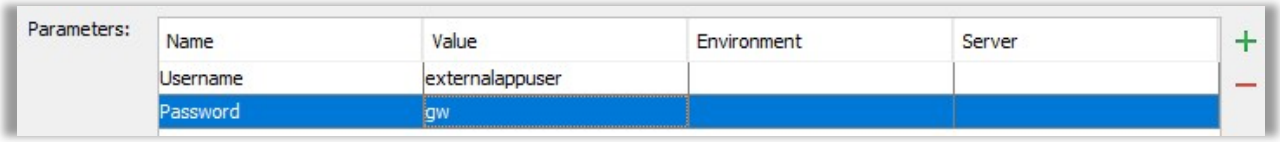
   - Add a Gosu plugin.



   - Configure the Gosu Class field.



   - Add plugin parameters.

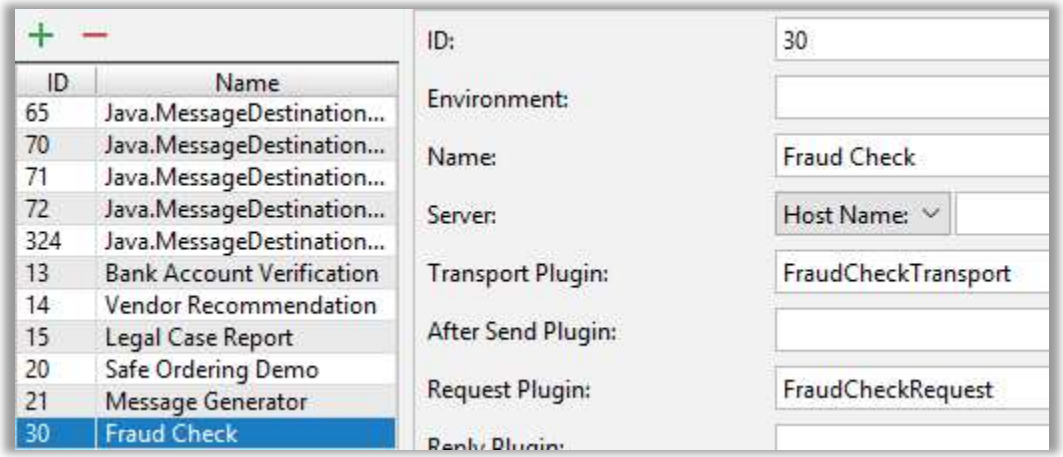


4. **Configure the destination.**

   a) Open **messaging-config.xml** file.

   b) Select **FraudCheck** destination.

   c) Add **FraudCheckTransport** in **Transport Plugin** field.



5. **Deploy code changes.**

   a) From the **Studio** menu, **Restart** the server.

6. **Perform verification steps.**