```
//EXAMPLE DATASTORESERVICE-CLASS
import com.sap.gateway.ip.core.customdev.util.Message

//Imports for DataStoreService-class
import com.sap.it.api.asdk.datastore.*
import com.sap.it.api.asdk.runtime.*

Message processData(Message message) {

        //Data to be stored in datatore
        def payload = "This is sample data"
        //Get service instance
        def service = new Factory(DataStoreService.class).getService()

        //Check if valid service instance was retrieved
        if( service != null) {
                def dBean = new DataBean()
                dBean.setDataAsArray(payload.getBytes("UTF-8"))

                //Define datatore name and entry id
                def dConfig = new DataConfig()
                dConfig.setStoreName("DatastoreFromGroovyASDK")
                dConfig.setId("TestEntry")
                dConfig.setOverwrite(true)

                //Write to data store
                result = service.put(dBean,dConfig)
        }
  return message
}
```

```
//EXAMPLE DATASTORE-CLASS
import com.sap.gateway.ip.core.customdev.util.Message

//Imports for the DataStore-class handling/access
import com.sap.esb.datastore.DataStore
import com.sap.esb.datastore.Data
import org.osgi.framework.*

Message processData(Message message) {

        //Get CamelContext and from that the DataStore instance
        def camelCtx = message.exchange.getContext()
        DataStore dataStore =
(DataStore)camelCtx.getRegistry().lookupByName(DataStore.class.getName())

        //Define headers and payload/body as byte[]
        Map<String, Object> headers = ["headerName1":"me", "anotherHeader": false]
        def payload = "This is sample data".getBytes("UTF-8")

        //Create datastore payload/data with the following parameters
        //params => (DatastoreName, ContextName, EntryId, Body, Headers, MessageId,
Version)
        //Note: Setting ContextName to null, will create a global Datastore
        Data dsData = new Data("DatastoreFromGroovyServiceImpl", null,
                    "TestEntry", payload, headers, "life-is-hard", 0)

        //Write dsData element to the data store
        //params => (DataInstance, overwriteEntry, encrypt, alertPeriodInMs, expirePeriodInMs)
        dataStore.put(dsData, true, false, 13824000000, 90552000000)

  return message
}
```