```
//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) {

    //Get service instance
    def service = new Factory(DataStoreService.class).getService()

    //Check if valid service instance was retrieved
    if( service != null) {
        //Read data store entry via id
        def dsEntry = service.get("DatastoreName","EntryId")
        def result = new String(dsEntry.getDataAsArray())
        message.setBody(body)
    }
    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) {
```

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


    //Read from Datastore params => (DatastoreName, EntryId)
    def dsEntry = dataStore.get("DatastoreFromGroovyServiceImpl", "TestEntry")
    //Get datastore entry payload as String
    def payload = new String(dsEntry.getDataAsArray())


    //NOTE: If you pass the name context-name ad third parameter, you can read from
    //      any datastore, regardless if it's global or local! The context name usually
    //      equals the id of the IFlow a Datastore belongs to.
    //def dsEntry = dataStore.get("DatastoreName", "ContextName", "EntryId")


    //If you want to select multiple entry, use select-method and pass the
    //following params => (DatastoreName , NumberOfEntriesToPull)
    //def dsEntryArray = dataStore.select("DatastoreName",  10)


    message.setBody(payload)
    return message
}
```