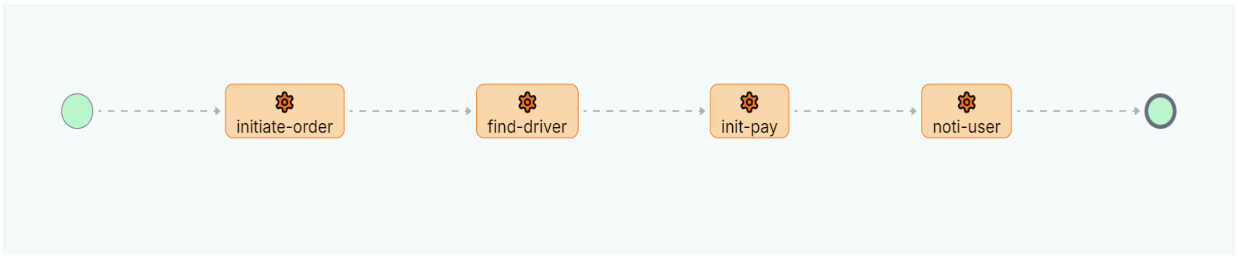
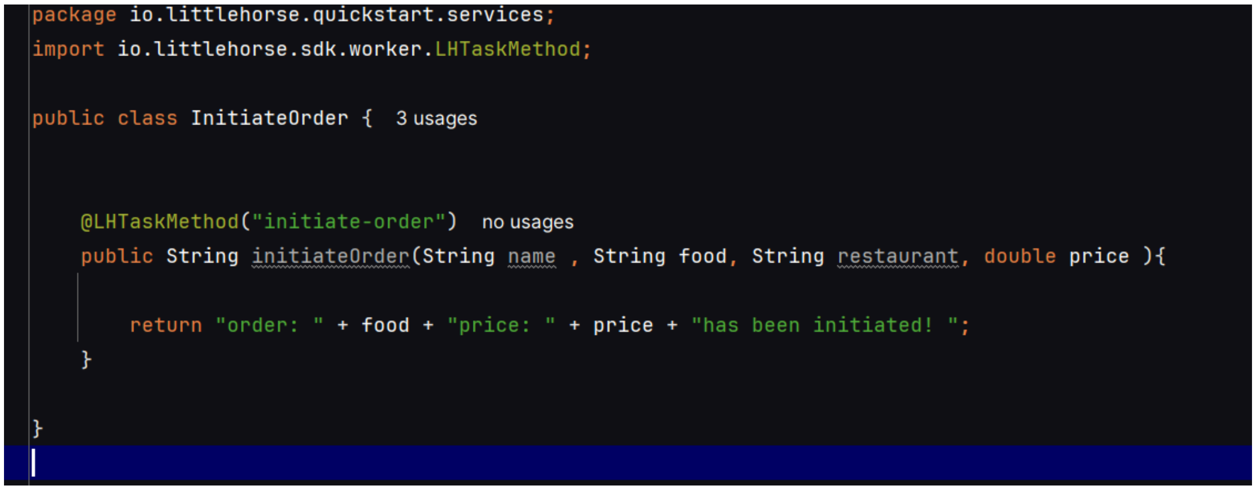
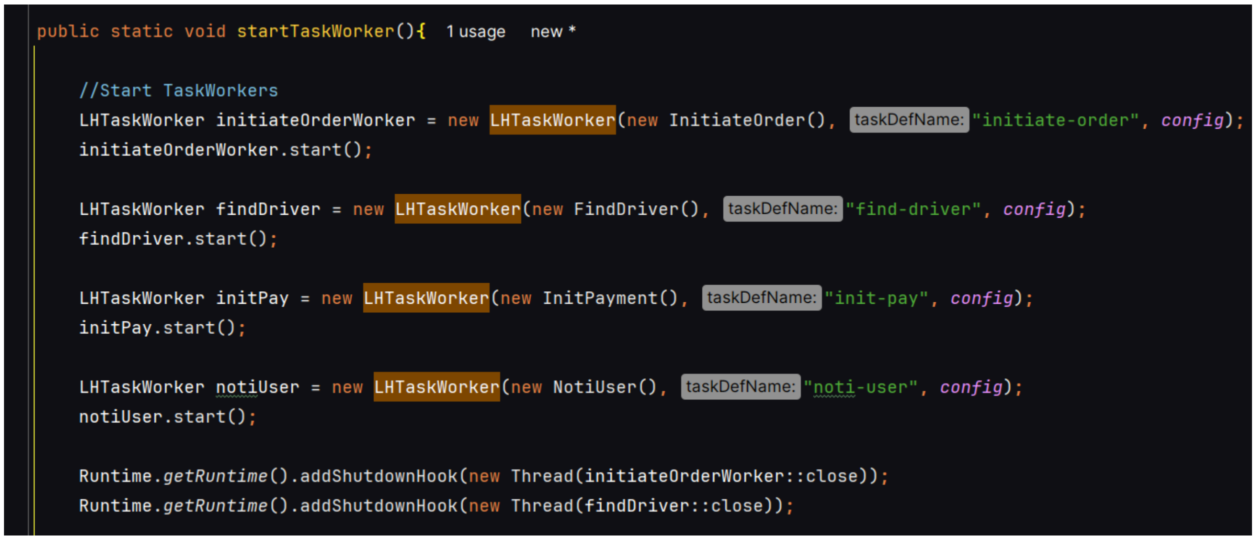
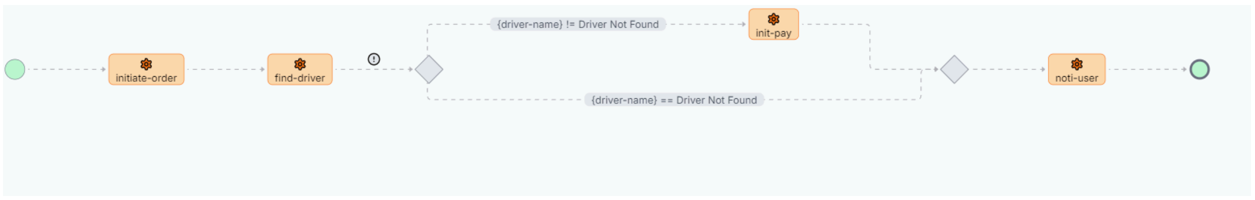




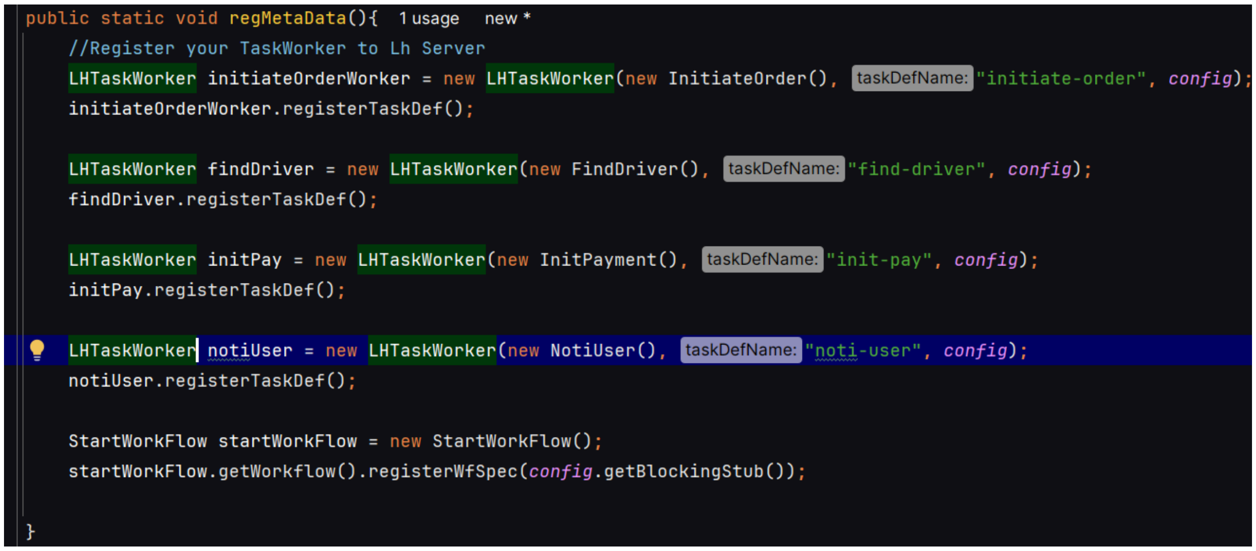
```
package io.littlehorse.quickstart.services;
import io.littlehorse.sdk.worker.LHTaskMethod;

public class InitiateOrder {  3 usages


    @LHTaskMethod("initiate-order")  no usages
    public String initiateOrder(String name , String food, String restaurant, double price ){

        return "order: " + food + "price: " + price + "has been initiated! ";
    }

}
```



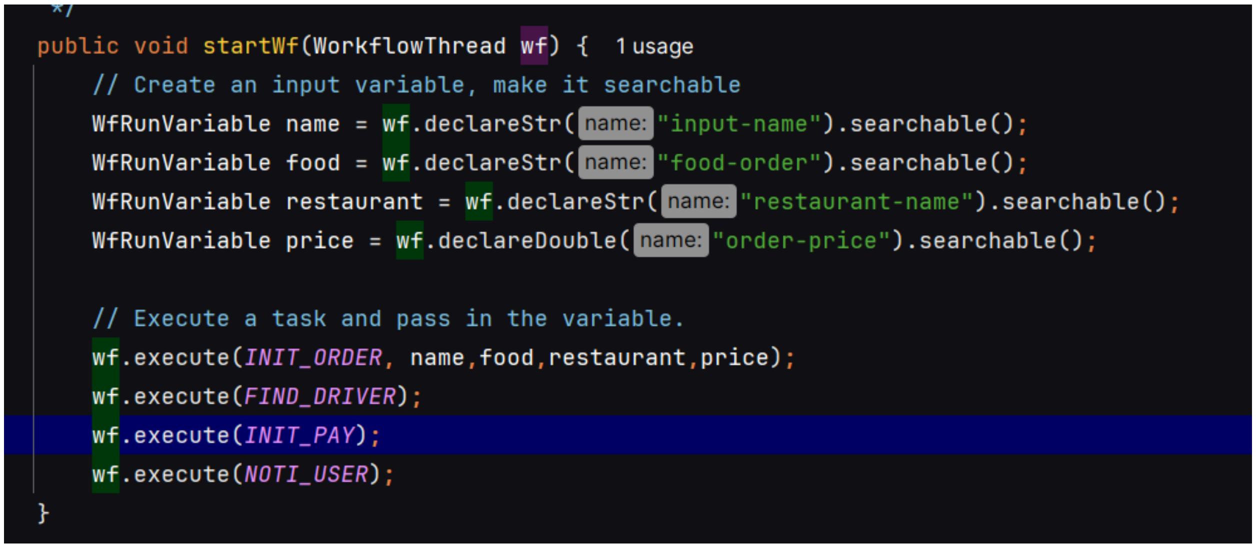
```
public static void startTaskWorker(){  1 usage   new *

    //Start TaskWorkers
    LHTaskWorker initiateOrderWorker = new LHTaskWorker(new InitiateOrder(), taskDefName: "initiate-order", config);
    initiateOrderWorker.start();

    LHTaskWorker findDriver = new LHTaskWorker(new FindDriver(), taskDefName: "find-driver", config);
    findDriver.start();

    LHTaskWorker initPay = new LHTaskWorker(new InitPayment(), taskDefName: "init-pay", config);
    initPay.start();

    LHTaskWorker notiUser = new LHTaskWorker(new NotiUser(), taskDefName: "noti-user", config);
    notiUser.start();

    Runtime.getRuntime().addShutdownHook(new Thread(initiateOrderWorker::close));
    Runtime.getRuntime().addShutdownHook(new Thread(findDriver::close));
```



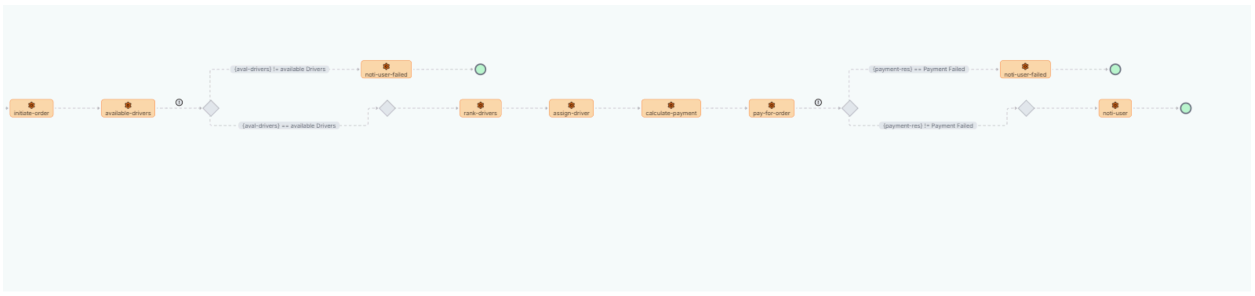
initiate-order
find-driver
{driver-name} != Driver Not Found
init-pay
{driver-name} == Driver Not Found
noti-user

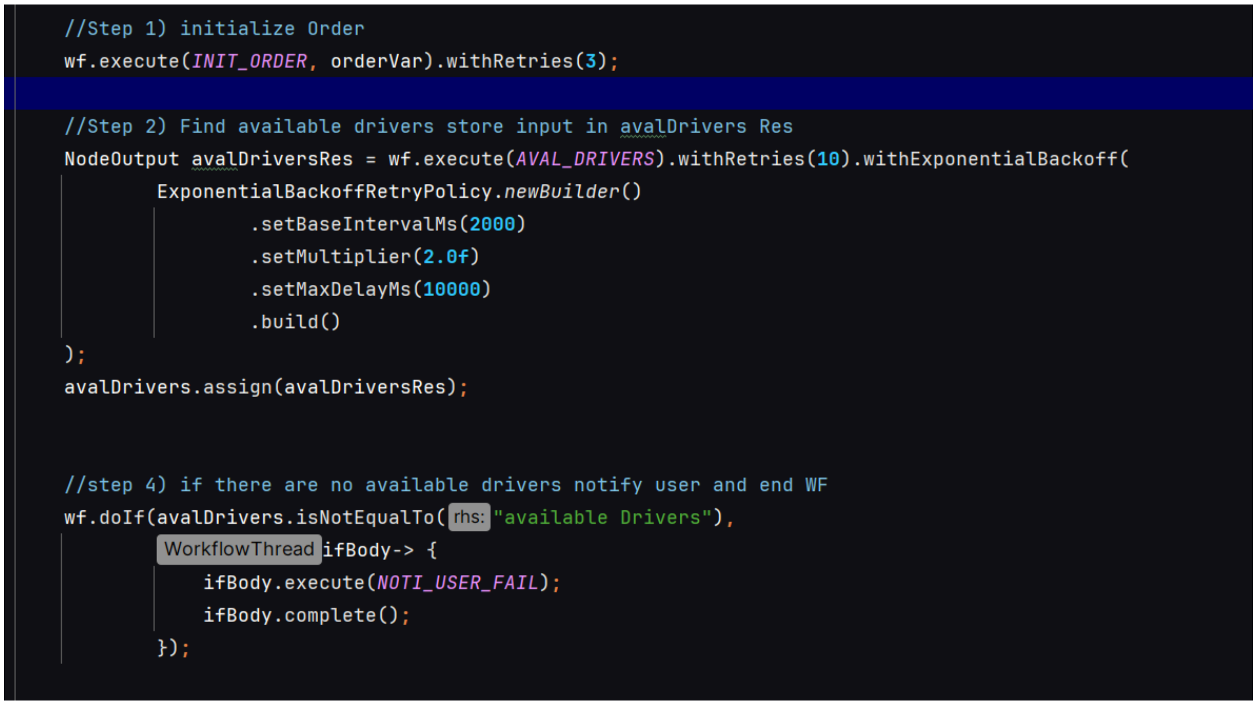


```java
public static void regMetaData(){  1 usage   new *
    //Register your TaskWorker to Lh Server
    LHTaskWorker initiateOrderWorker = new LHTaskWorker(new InitiateOrder(), taskDefName: "initiate-order", config);
    initiateOrderWorker.registerTaskDef();

    LHTaskWorker findDriver = new LHTaskWorker(new FindDriver(), taskDefName: "find-driver", config);
    findDriver.registerTaskDef();

    LHTaskWorker initPay = new LHTaskWorker(new InitPayment(), taskDefName: "init-pay", config);
    initPay.registerTaskDef();

    LHTaskWorker notiUser = new LHTaskWorker(new NotiUser(), taskDefName: "noti-user", config);
    notiUser.registerTaskDef();

    StartWorkFlow startWorkFlow = new StartWorkFlow();
    startWorkFlow.getWorkflow().registerWfSpec(config.getBlockingStub());

}
```

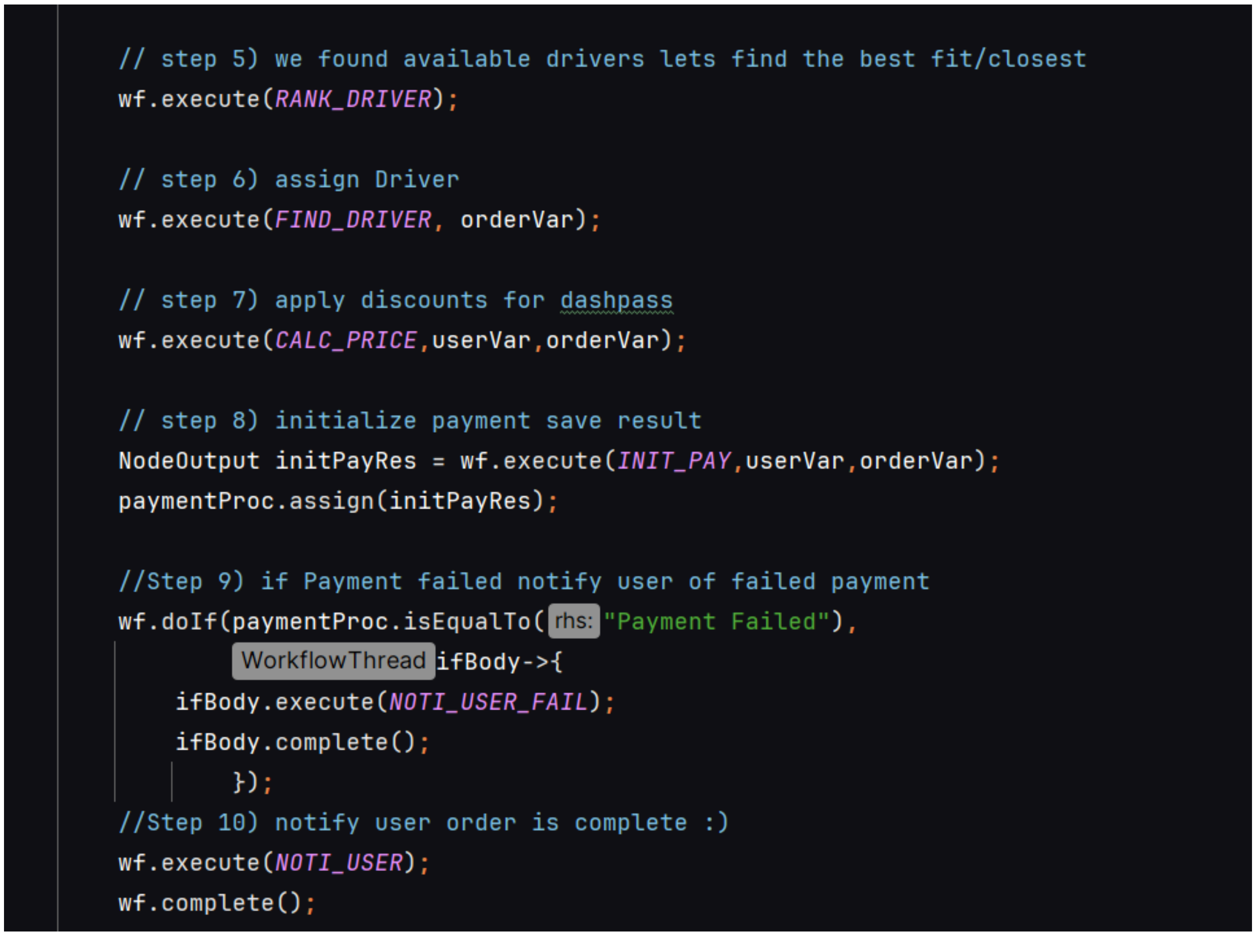


```java
 */
public void startWf(WorkflowThread wf) {  1 usage
    // Create an input variable, make it searchable
    WfRunVariable name = wf.declareStr( name: "input-name").searchable();
    WfRunVariable food = wf.declareStr( name: "food-order").searchable();
    WfRunVariable restaurant = wf.declareStr( name: "restaurant-name").searchable();
    WfRunVariable price = wf.declareDouble( name: "order-price").searchable();

    // Execute a task and pass in the variable.
    wf.execute(INIT_ORDER, name,food,restaurant,price);
    wf.execute(FIND_DRIVER);
    wf.execute(INIT_PAY);
    wf.execute(NOTI_USER);
}
```


initiate-order
available-drivers
{aval-drivers} != available Drivers
noti-user-failed
{aval-drivers} == available Drivers
rank-drivers
assign-driver
calculate-payment
pay-for-order
{payment-res} == Payment Failed
noti-user-failed
{payment-res} != Payment Failed
noti-user



```
//Step 1) initialize Order
wf.execute(INIT_ORDER, orderVar).withRetries(3);

//Step 2) Find available drivers store input in avalDrivers Res
NodeOutput avalDriversRes = wf.execute(AVAL_DRIVERS).withRetries(10).withExponentialBackoff(
        ExponentialBackoffRetryPolicy.newBuilder()
                .setBaseIntervalMs(2000)
                .setMultiplier(2.0f)
                .setMaxDelayMs(10000)
                .build()
);
avalDrivers.assign(avalDriversRes);


//step 4) if there are no available drivers notify user and end WF
wf.doIf(avalDrivers.isNotEqualTo( rhs: "available Drivers"),
        WorkflowThread ifBody-> {
            ifBody.execute(NOTI_USER_FAIL);
            ifBody.complete();
        });
```



```
// step 5) we found available drivers lets find the best fit/closest
wf.execute(RANK_DRIVER);

// step 6) assign Driver
wf.execute(FIND_DRIVER, orderVar);

// step 7) apply discounts for dashpass
wf.execute(CALC_PRICE,userVar,orderVar);

// step 8) initialize payment save result
NodeOutput initPayRes = wf.execute(INIT_PAY,userVar,orderVar);
paymentProc.assign(initPayRes);

//Step 9) if Payment failed notify user of failed payment
wf.doIf(paymentProc.isEqualTo( rhs: "Payment Failed"),
        WorkflowThread ifBody->{
    ifBody.execute(NOTI_USER_FAIL);
    ifBody.complete();
        });
//Step 10) notify user order is complete :)
wf.execute(NOTI_USER);
wf.complete();
```