Gunit

# Builder Output:

gosu  
package gw.api.databuilder  
  
uses com.ibm.icu.util.CurrencyAmount  
uses java.lang.Integer  
uses java.util.Date  
uses java.math.BigDecimal  
uses gw.api.locale.DisplayKey  
uses gw.api.util.CurrencyUtil  
uses gw.api.database.Query  
uses gw.api.databuilder.BuilderContext  
uses gw.api.database.spatial.SpatialPoint  
  
@Export  
class VehicleBuilder extends gw.api.databuilder.CCDataBuilder<Vehicle, VehicleBuilder> {  
  
 construct() {  
 super(Vehicle)  
 }  
  
 public function withBoatType(boatType : BoatType) : VehicleBuilder {  
 set(Vehicle.Type.TypeInfo.getProperty("BoatType"), boatType)  
 return this  
 }  
  
 public function withColor(color : String) : VehicleBuilder {  
 set(Vehicle.Type.TypeInfo.getProperty("Color"), color)  
 return this  
 }  
  
 public function withLicensePlate(licensePlate : String) : VehicleBuilder {  
 set(Vehicle.Type.TypeInfo.getProperty("LicensePlate"), licensePlate)  
 return this  
 }  
  
 public function withLoan(loan : Boolean) : VehicleBuilder {  
 set(Vehicle.Type.TypeInfo.getProperty("Loan"), loan)  
 return this  
 }  
  
 public function withLoanMonthlyPayment(loanMonthlyPayment : CurrencyAmount) : VehicleBuilder {  
 set(Vehicle.Type.TypeInfo.getProperty("LoanMonthlyPayment"), loanMonthlyPayment)  
 return this  
 }  
  
 public function withLoanMonthsRemaining(loanMonthsRemaining : Integer) : VehicleBuilder {  
 set(Vehicle.Type.TypeInfo.getProperty("LoanMonthsRemaining"), loanMonthsRemaining)  
 return this  
 }  
  
 public function withLoanPayoffAmount(loanPayoffAmount : CurrencyAmount) : VehicleBuilder {  
 set(Vehicle.Type.TypeInfo.getProperty("LoanPayoffAmount"), loanPayoffAmount)  
 return this  
 }  
  
 public function withMake(make : String) : VehicleBuilder {  
 set(Vehicle.Type.TypeInfo.getProperty("Make"), make)  
 return this  
 }  
  
 public function withManufacturer(manufacturer : VehicleManufacturer) : VehicleBuilder {  
 set(Vehicle.Type.TypeInfo.getProperty("Manufacturer"), manufacturer)  
 return this  
 }  
  
 public function withModel(model : String) : VehicleBuilder {  
 set(Vehicle.Type.TypeInfo.getProperty("Model"), model)  
 return this  
 }  
  
 public function withOffRoadStyle(offRoadStyle : OffRoadVehicleStyle) : VehicleBuilder {  
 set(Vehicle.Type.TypeInfo.getProperty("OffRoadStyle"), offRoadStyle)  
 return this  
 }  
  
 public function withSerialNumber(serialNumber : String) : VehicleBuilder {  
 set(Vehicle.Type.TypeInfo.getProperty("SerialNumber"), serialNumber)  
 return this  
 }  
  
 public function withState(state : Jurisdiction) : VehicleBuilder {  
 set(Vehicle.Type.TypeInfo.getProperty("State"), state)  
 return this  
 }  
  
 public function withStyle(style : VehicleStyle) : VehicleBuilder {  
 set(Vehicle.Type.TypeInfo.getProperty("Style"), style)  
 return this  
 }  
  
 public function withTestVehicleNum\_Ext(testVehicleNum\_Ext : Integer) : VehicleBuilder {  
 set(Vehicle.Type.TypeInfo.getProperty("TestVehicleNum\_Ext"), testVehicleNum\_Ext)  
 return this  
 }  
  
 public function withVin(vin : String) : VehicleBuilder {  
 set(Vehicle.Type.TypeInfo.getProperty("Vin"), vin)  
 return this  
 }  
  
 public function withYear(year : Integer) : VehicleBuilder {  
 set(Vehicle.Type.TypeInfo.getProperty("Year"), year)  
 return this  
 }  
  
 public function withLienholders(lienholder : VehicleOwner) : VehicleBuilder {  
 addArrayElement(Vehicle.Type.TypeInfo.getProperty("Lienholders"), lienholder)  
 return this  
 }  
}

# Gunit Output:

gosu  
uses gw.api.util.CCCurrencyUtil  
uses gw.api.util.CurrencyUtil  
uses gw.sampledata.SampleDataGroup  
uses gw.suites.CCExampleServerSuite  
uses gw.api.test.CCServerTestClassBase  
uses gw.testharness.v3.Suites  
uses gw.api.util.DateUtil  
uses gw.api.databuilder.\*  
uses java.util.Date  
uses java.math.BigDecimal  
uses gw.api.locale.DisplayKey  
uses gw.api.database.Query  
uses gw.api.database.spatial.SpatialPoint  
  
class ChangeVehicleYearTest extends CCServerTestClassBase {  
  
 construct() { }  
  
 function testChangeVehicleYear() {  
 // Given  
 var vehicle = createVehicle()  
 var originalYear = vehicle.Year  
  
 // When  
 changeVehicleYear(vehicle)  
  
 // Then  
 assertThat(vehicle.Year).isEqualTo(1990)  
 assertThat(vehicle.Year).isNotEqualTo(originalYear)  
 }  
  
 public static function changeVehicleYear(vehicle: Vehicle) {  
 if(vehicle != null) {  
 vehicle.Year = 1990  
 }  
 }  
  
 function createVehicle() : Vehicle {  
 var vehicleBuilder = new VehicleBuilder()  
 .withBoatType(typekey.BoatType.TC\_CO)  
 .withColor("Red")  
 .withCreateTime(Date.Now)  
 .withLicensePlate("ABC201")  
 .withLoan(true)  
 .withLoanMonthsRemaining(36)  
 .withMake("Tata")  
 .withManufacturer(VehicleManufacturer.TC\_ACUR)  
 .withModel("Nexon")  
 .withOffRoadStyle(OffRoadVehicleStyle.TC\_SKT)  
 .withSerialNumber("serial201")  
 .withState(Jurisdiction.TC\_AZ)  
 .withStyle(VehicleStyle.TC\_BOAT)  
 .withUpdateTime(Date.Now)  
 .withVin("VIN201")  
 .withYear(2021)  
  
 return vehicleBuilder.create()  
 }  
}