End to End Quote Guide

The purpose of this user guide is to show how to create an entire Quote using the *QuoteParam*.

Table of Contents

[End to End Quote Guide 1](#_Toc85617047)

[Table of Contents 1](#_Toc85617048)

[Instructions 2](#_Toc85617049)

[QuoteParam 2](#_Toc85617050)

Instructions

# QuoteParam

The *QuoteParam* is the main class that define a Quote, until now we can define Vehicles, Drivers, Policy Coverages (Limit) and Modifiers.

## **Initilialization**

To initialize or create a new ***QuoteParam*** we need to specify the ***state*** and ***algorithmCode*** or ***state*** and ***classCodeKeyword****.*

**Example** with *state* and *algorithmCode*:

var state = "IL";

var algorithmCode = "VA00029";

var quoteParam = new QuoteParam(state, algorithmCode);

**Example** with *state* and *classCodeKeyword*:

var state = "IL";

var classCodeKeyword = ClassCodeKeyword.GetUsingKeywordName("Towing Services");

var quoteParam = new QuoteParam(state, classCodeKeyword);

When you create a new *QuoteParam* by default the following properties are filled:

* **State**: filled by the *state* passed by parameter
* **ClassCodeKeyword**: filled by the c*lassCodeKeyword* passed by parameter or based in the *algorithCode* passed by parameter
* **LimitidParam**.**Limits:** It added the coverage based on the *classCodeKeyword* if has one otherwise the Coverage Type *BIPD* it’s added by default.
* **Organization**: create an *Organization* based on the *classCodeKeyword*
* **RatableObjectEffectiveDate**: the tomorrow date. For example: if today is 01/01/2021, the *RatableObjectEffectiveDate* will have the value of 01/02/2021
* **RatableObjectExpirationDate**: one year and one day since today date*.* For example: if today is 01/01/2021, *RatableObjectExpirationDate* will have the value of 01/02/2022

All these properties can be modified.

## **Param Properties**

### **VehicleParam**

When a *QuoteParam* is created, by default, a new vehicle is added to the Quote. You can modify the defaults values, also you can add a new one:

var state = "IL";

var classCodeKeyword = ClassCodeKeyword.GetUsingKeywordName("Towing Services");

var quoteParam = new QuoteParam(state, classCodeKeyword);

// Modify defaults VehicleParam properties

quoteParam.VehicleParam.Vehicles[0].Object.make = "Ford";

quoteParam.VehicleParam.Vehicles[0].Object.model = "Explorer";

quoteParam.VehicleParam.Vehicles[0].Object.yearOfManufacture = 2019;

quoteParam.VehicleParam.Vehicles[0].Object.antitheft = true;

quoteParam.VehicleParam.Vehicles[0].VehicleQuentionAnswerParam.TransportTours.\_response = "false";

// Create a new Vehicle

var newVehicleParam = new VehicleParam();

newVehicleParam.Object.make = "Honda";

newVehicleParam.Object.model = "Accord";

newVehicleParam.Object.yearOfManufacture = 2012;

// Add the new Vehicle to a quote

quoteParam.VehicleParam.Vehicles.Add(newVehicleParam);

### **DriverParam**

When a *QuoteParam* is created, by default, a new driver is added to the Quote. You can modify the defaults values, also you can add a new one:

var state = "IL";

var classCodeKeyword = ClassCodeKeyword.GetUsingKeywordName("Towing Services");

var quoteParam = new QuoteParam(state, classCodeKeyword);

// Modify defaults DriverParam properties

quoteParam.DriverParam.Drivers[0].Object.dateOfBirth = new DateTime(1989, 10, 10);

quoteParam.DriverParam.Drivers[0].Object.person.primaryName.given = "John";

quoteParam.DriverParam.Drivers[0].DriverQuentionAnswerParam.ExcludeDriver.\_response = "false";

// Create a new Driver

var newDriverParam = new DriverParam();

newDriverParam.Object.dateOfBirth = new DateTime(1989, 10, 10);

newDriverParam.Object.person.primaryName.given = "Jane";

// Add the new Driver to a quote

quoteParam.DriverParam.Drivers.Add(newDriverParam);

### **LimitParam (Policy Coverage)**

When a *QuoteParam* is created, by default, a new Coverage Type is added to the Quote based in the *classCodeKeyword* classCodeKeyword if has one otherwise the Coverage Type BIPD it’s added by default.

You can add a new one:

var state = "IL";

var classCodeKeyword = ClassCodeKeyword.GetUsingKeywordName("Towing Services");

var quoteParam = new QuoteParam(state, classCodeKeyword);

// Add Coverage

quoteParam.LimitParam.Add(CoverageType.IN\_TOW);

quoteParam.LimitParam.Add(CoverageType.COLLISION);

### **ModifierParam**

You can add *Experience Rating* or *Schedule Rating* modifiers after you create a QuoteParam

var state = "IL";

var classCodeKeyword = ClassCodeKeyword.GetUsingKeywordName("Towing Services");

var quoteParam = new QuoteParam(state, classCodeKeyword);

// Add Schedule Rating Modifier

quoteParam.ModifierParam.MANAGEMENT = 2;

quoteParam.ModifierParam.EMPLOYEES = 10;

quoteParam.ModifierParam.EQUIPMENT = 3.5m;

// Add Experience Rating Modifier

quoteParam.ModifierParam.ExperienceRating = 3.5m;

After creating the QuoteParam as you want, send it to the API. This will return an Entity Quote from the server.

var entityQuote = quoteParam.RunThisThroughAPI();