Rolls

het verschil in vergelijken

Authentication Instruction

Version: 31-05-2022

History

31-05-2022	First English version

Introduction

In order to be able to work with our vehicle service, authentication is needed. Authentication will be explained in this document based on our function ‘basic license plate’.

The function ‘basic license plate’ returns basic vehicle information, including a link to a picture. The format is in JSON or XML.

Example of the response

Below you will find an example of a response xml, where no explanation is needed for nodes.

```
<?xml version="1.0" encoding="utf-16"?>
<LicenseplatesController.BasicVehicleData xmlns:i="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://schemas.kooijmans.nl/vehicledata">
  <BrandLogo>108.jpg</BrandLogo>
  <BrandLogoThumb>Thumb/108.jpg</BrandLogoThumb>
  <ConstructionMonth>8</ConstructionMonth>
  <ConstructionYear>2011</ConstructionYear>
  <ConsumerPrice>16770</ConsumerPrice>
  <CurrentValue>4696</CurrentValue>
  <FuelTypeDescription>Benzine</FuelTypeDescription>
  <Make>Fiat</Make>
  <Model>500 </Model>
  <PathLogo>https://autofotos.autotelexpro.nl/AutoFotos/Merklogos/</PathLogo>
  <PathPhoto>https://autofotos.autotelexpro.nl/AutoFotos/</PathPhoto>
  <PhotoFront>FIAT/ATX_M671_F1.JPG</PhotoFront>
  <PhotoFrontThumb>FIAT/Thumb/ATX_M671_F1.JPG</PhotoFrontThumb>
  <PhotoInterior>FIAT/ATX_M671_I1.JPG</PhotoInterior>
  <PhotoInteriorThumb>FIAT/Thumb/ATX_M671_I1.JPG</PhotoInteriorThumb>
  <PhotoRear>FIAT/ATX_M671_R1.JPG</PhotoRear>
  <PhotoRearThumb>FIAT/Thumb/ATX_M671_R1.JPG</PhotoRearThumb>
  <SecuritySystemClass>1</SecuritySystemClass>
  <Source>Autotelex+RDW</Source>
  <TypeDescriptionLong>0.9 Twinair Lounge</TypeDescriptionLong>
  <VehicleType>PassengerCar</VehicleType>
</LicenseplatesController.BasicVehicleData>
```

Authentication

In order to invoke the functions of our service, you must use these HTTP headers:

headers.Add($"ClientID:{clientID}");
headers.Add($"ClientSecret:{clientSecret}");

Your clientID and clientSecret identify your account and will be provided in a separate mail.

Application to be used by multiple insurance agents

It is also necessary to provide the RollsEngineLicenseKey of the insurance agent in case your application is to be used by multiple insurance agents. It enables us to identify the right agent and charge them for their usage (if applicable). This RollsEngineLicenseKey (GUID) can be supplied by our helpdesk. It is preferred to create a setting in your software, so that the insurance agent can request and enter it themselves.

It is necessary to provide RollsEngineLicenseKey in the headers as well:

headers.Add($"RollsEngineLicenseKey:{rollsEngineLicenseKey}");

It is possible for us to connect the RollsEngineLicenseKey to your ClientID in our administration in case all calls will be paid by one party, and if it is not expected that other parties will use the same application. In that case you won't need to provide this header.

Availability

Vehicles.rolls.nl is not a static location!
On behalf of our failover-system, our services operate on multiple locations.
Our DNS-service returns only the IP-address of a healthy location and
manages a TTL of 30 seconds (validity of the IP-address).

In .Net you can take care of the DNS conversion taking place every call by using:
System.Net.ServicePointManager.DnsRefreshTimeout = 0;

In order to use our failover-system in full, your client-component should not use caching or sliding expiration. For testing purposes, you will change a DNS yourself, while the call will be executed on repeat. As soon as the TTL is over, a new location should be used.

Unique feature for traceability

For support reasons a fast way of tracing a request is required. Therefore we offer a traceid which can be added to the querystring. The value of the traceid is up to you, the only requirement is that it produces a valid URL. However, we advise to use a GUID. Adding the name of your server is optional.

It is important that every call, even a repeated call, is supplied with a unique traceid. In case you deliver a specific querystring of a request that gave problems, we can trace that request quickly.

Examples of Requests with traceid:

https://vehicles.rolls.nl/api/licenseplates/basic/38RTP9?traceid ={80168F2E-6570-4CA3-B9EE-DEF2C84354C5}

https://vehicles.rolls.nl/api/licenseplates/basic/38RTP9?traceid ={80168F2E-6570-4CA3-B9EE-DEF2C84354C5-Server1}

Unknown license plate

In restful WebApi Service it is common to return a 404 (not found) error if no results are available. This does not indicate that our service is out of reach. It indicates that the specific license plate is not available to a vehicle in our database. You can test this by making a request for license plate YB-98-VH.

Caching vehicle data

Our databases with vehicle data are updated daily. You can cache requested vehicle data on the same day, so that you don't need to make a new call once the same data is requested again on the same day.

Building in .NET

In this example the vehicle service is requested:

```
public async static Task<string> CallVehicleServiceAsync(
                                                        string clientID,
                                                        string clientSecret,
                                                        string rollsEngineLicenseKey = ""
        )
    {
        string content = "";
        try
        {
            string webApiUri = "https://vehicles.rolls.nl";
            string requestUri = "/api/licenseplates/basic/38RTP9";

            System.Net.ServicePointManager.DnsRefreshTimeout = 0;

           //  For securityprotocol TLS1.2 in .NET Framework <= 4.5 the securityprotocol must
           //  be specified. This line is NOT required in .NET Framework 4.6 or newer versions.
           //System.Net.ServicePointManager.SecurityProtocol = System.Net.SecurityProtocolType.Tls12;

            System.Net.Http.HttpClient client = new System.Net.Http.HttpClient
            {
                BaseAddress = new Uri(webApiUri)
            };

            client.DefaultRequestHeaders.Add("ClientID", clientID);
            client.DefaultRequestHeaders.Add("ClientSecret", clientSecret);

            //  If you develop the vehiclecall for other/multiple rolls-customers who pay for
            //  each call themselves, you must pass their rollsEngineLicenseKey. Access for
            //  that customer must be granted by us, so please contact us.
            if(rollsEngineLicenseKey != "")
                client.DefaultRequestHeaders.Add("RollsEngineLicenseKey", rollsEngineLicenseKey);

            //  If result is required in Dutch uncomment this line.
            //client.DefaultRequestHeaders.Add("Accept-Language", "nl-NL");

            //  JSON is default, if resultformat XML is required, uncomment this line.
            //client.DefaultRequestHeaders.Add("Accept", "text/xml");

            var res = await client.GetAsync(requestUri).ConfigureAwait(false);
            res.EnsureSuccessStatusCode();
            content = await res.Content.ReadAsStringAsync().ConfigureAwait(false);
        }
        catch (Exception e)
        {
            content = e.Message;
        }

        return content;
    }
```

Buillding in PHP

In this example the vehicle service is requested:

```
<Html>
	<head>
		<title>Php call vehicle service sample</title>
	</head>
	<body>
		<?php

		$licenseplate='31-XG-PX';
		// TODO: Update these values with your credentials
		$clientID = 'PLACE_YOUR_CLIENT_ID_HERE';
		$clientSecret = 'PLACE_YOUR_CLIENT_SECRET_HERE';
		// If you develop the vehiclecall for other/multiple rolls-customers who pay for
		// each call themselves, you must pass their rollsEngineLicenseKey. Access for
		// that customer must be granted by us, so please contact us.
		//$rollsEngineLicenseKey = '<GUID supplied by our helpdesk>';

		$dateNowUTC = new DateTime(null, new DateTimeZone('UTC'));

		$acceptLanguageHeader = 'Accept-Language: nl-NL';
		$acceptHeader = 'Accept: text/xml';

		$clientIDHeader = 'ClientID: ' . $clientID;
		$clientSecretHeader = 'ClientSecret: ' . $clientSecret;
		//RollsEngineLicenseKey is sometimes also is necessary. If left out and no exception occurs you don't need it with your ClientID.
		//$rollsEngineLicenseKeyHeader = 'RollsEngineLicenseKey: ' . $rollsEngineLicenseKey;

		//XML
		print ('<HR/>');
		print '<b>XML result:</b><br/>';

		$CurlXml = curl_init( 'https://vehicles.rolls.nl/api/licenseplates/basic/' . $licenseplate);
		curl_setopt( $CurlXml, CURLOPT_HTTPHEADER, array($clientIDHeader, $clientSecretHeader, $acceptLanguageHeader, $acceptHeader));
		curl_setopt( $CurlXml, CURLOPT_SSL_VERIFYPEER, false);
		curl_setopt( $CurlXml, CURLOPT_RETURNTRANSFER, 1);

		$XmlResult = curl_exec( $CurlXml );
		$xml = new SimpleXMLElement($XmlResult);
		$xml->registerXPathNamespace('ks', 'http://schemas.rolls.nl/vehicledata');
		$MakeNames = $xml->xpath('//ks:Make');

		print ('<textarea rows="15" cols="100">');
		print ($XmlResult);
		print ('</textarea>');
		print '<br/>Inhoud van de node met xpath "//Make":';
		print ($MakeNames[0]);

		//JSON
		print ('<HR/>');
		print '<b>JSON result:(default)</b><br/>';
		$CurlJson = curl_init( 'https://vehicles.rolls.nl/api/licenseplates/basic/' . $licenseplate);
		curl_setopt( $CurlJson, CURLOPT_HTTPHEADER, array($clientIDHeader, $clientSecretHeader, $acceptLanguageHeader));
		curl_setopt( $CurlJson, CURLOPT_SSL_VERIFYPEER, false);
		curl_setopt( $CurlJson, CURLOPT_RETURNTRANSFER, 1);
		$jsonResult = curl_exec( $CurlJson );
		$json = json_decode($jsonResult);

		print ('<textarea rows="10" cols="100">');
		print ($jsonResult);
		print ('</textarea>');
		print ('<br/>Inhoud van de json property "Make":');
		print ($json->{'Make'});
		?>
	</body>
</Html>
```