Admin Service

The Administration Service provides administrative operations

Click here for the Service Description WSDL and Annotated XSDs

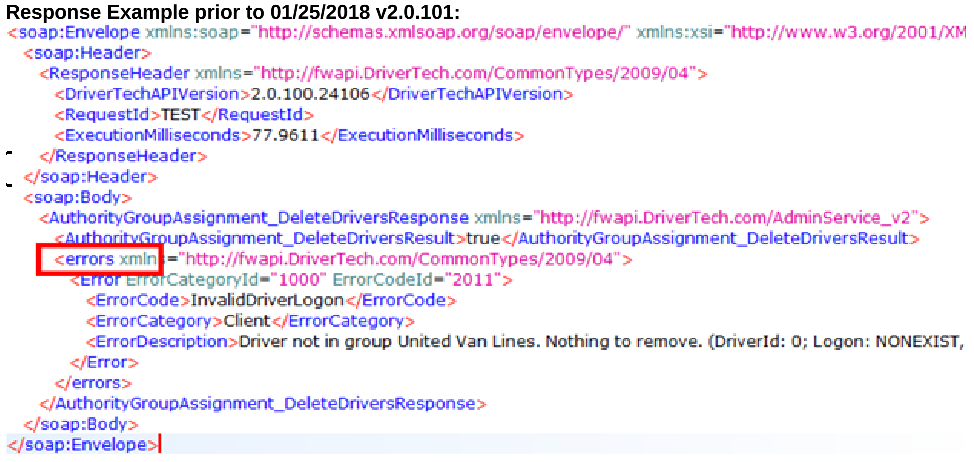
Recent Updates:

- **ENHANCED in v2.0.106 - 03/24/2022**
  - **Driver First and Last name maximum length expanded to 30/31 chars respectively**
  - **Driver FullName maximum length expanded to 61chars**

- **NEW FEATURES in v2.0.102 - 03/10/2018**
  - **Settings Group Membership Assignment for Vehicles on Vehicle, Vehicle Type and Terminal**

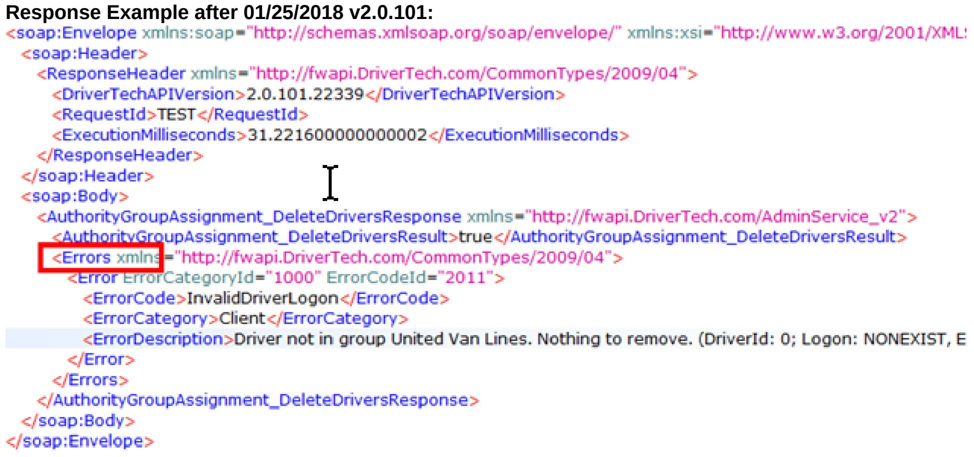
- **CORRECTION in v2.0.101 - 01/25/2018 - PLEASE UPDATE YOUR IMPLEMENTATION**
  - AuthorityGroupAssignment operation response ‘errors’ element has been changed to “Errors” due to issues it caused with newer java wsdlimport

    Response Example prior to 01/25/2018 v2.0.101:



```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XM
  <soap:Header>
    <ResponseHeader xmlns="http://fwapi.DriverTech.com/CommonTypes/2009/04">
      <DriverTechAPIVersion>2.0.100.24106</DriverTechAPIVersion>
      <RequestId>TEST</RequestId>
      <ExecutionMilliseconds>77.9611</ExecutionMilliseconds>
    </ResponseHeader>
  </soap:Header>
  <soap:Body>
    <AuthorityGroupAssignment_DeleteDriversResponse xmlns="http://fwapi.DriverTech.com/AdminService_v2">
      <AuthorityGroupAssignment_DeleteDriversResult>true</AuthorityGroupAssignment_DeleteDriversResult>
      <errors xmlns="http://fwapi.DriverTech.com/CommonTypes/2009/04">
        <Error ErrorCategoryId="1000" ErrorCodeId="2011">
          <ErrorCode>InvalidDriverLogon</ErrorCode>
          <ErrorCategory>Client</ErrorCategory>
          <ErrorDescription>Driver not in group United Van Lines. Nothing to remove. (DriverId: 0; Logon: NONEXIST,
        </Error>
      </errors>
    </AuthorityGroupAssignment_DeleteDriversResponse>
  </soap:Body>
</soap:Envelope>
```

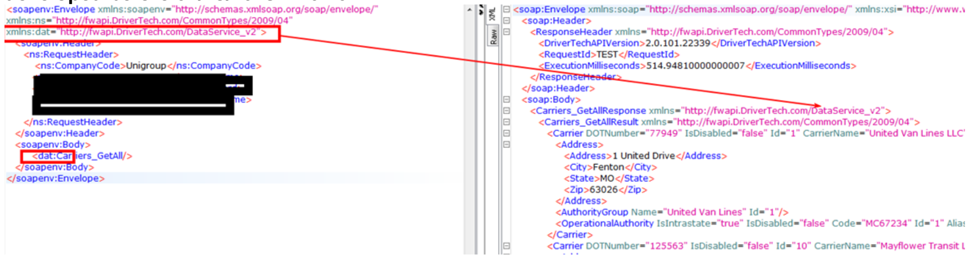
    Response Example after 01/25/2018 v2.0.101:



```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w3.org/2001/XML
  <soap:Header>
    <ResponseHeader xmlns="http://fwapi.DriverTech.com/CommonTypes/2009/04">
      <DriverTechAPIVersion>2.0.101.22339</DriverTechAPIVersion>
      <RequestId>TEST</RequestId>
      <ExecutionMilliseconds>31.221600000000002</ExecutionMilliseconds>
    </ResponseHeader>
  </soap:Header>
  <soap:Body>
    <AuthorityGroupAssignment_DeleteDriversResponse xmlns="http://fwapi.DriverTech.com/AdminService_v2">
      <AuthorityGroupAssignment_DeleteDriversResult>true</AuthorityGroupAssignment_DeleteDriversResult>
      <Errors xmlns="http://fwapi.DriverTech.com/CommonTypes/2009/04">
        <Error ErrorCategoryId="1000" ErrorCodeId="2011">
          <ErrorCode>InvalidDriverLogon</ErrorCode>
          <ErrorCategory>Client</ErrorCategory>
          <ErrorDescription>Driver not in group United Van Lines. Nothing to remove. (DriverId: 0; Logon: NONEXIST, E
        </Error>
      </Errors>
    </AuthorityGroupAssignment_DeleteDriversResponse>
  </soap:Body>
</soap:Envelope>
```

  - Carrier_GetAll default namespace has been changed to AdminService_v2
    - The service will detect if the client request uses DataService_v2 and will rewrite the response to use that namespace to prevent breaking change. Otherwise, the service will respond with the correct NS of AdminService_v2

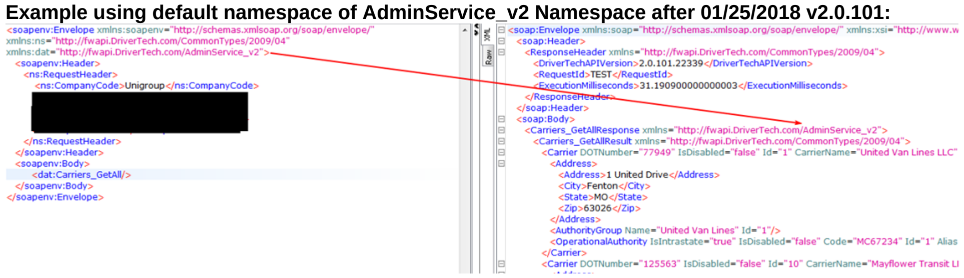
      Example using incorrect DataService_v2 Namespace - Service will rewrite response to match to prevent breaking client integrations developed before 01/25/2018 v2.0.101:



```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns="http://fwapi.DriverTech.com/CommonTypes/2009/04"
xmlns:dat="http://fwapi.DriverTech.com/DataService_v2">
  <soapenv:Header>
    <ns:RequestHeader>
      <ns:CompanyCode>Unigroup</ns:CompanyCode>
      [redacted]
    </ns:RequestHeader>
  </soapenv:Header>
  <soapenv:Body>
    <dat:Carriers_GetAll/>
  </soapenv:Body>
</soapenv:Envelope>
```

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.v
  <soap:Header>
    <ResponseHeader xmlns="http://fwapi.DriverTech.com/CommonTypes/2009/04">
      <DriverTechAPIVersion>2.0.101.22339</DriverTechAPIVersion>
      <RequestId>TEST</RequestId>
      <ExecutionMilliseconds>514.94810000000007</ExecutionMilliseconds>
    </ResponseHeader>
  </soap:Header>
  <soap:Body>
    <Carriers_GetAllResponse xmlns="http://fwapi.DriverTech.com/DataService_v2">
      <Carriers_GetAllResult xmlns="http://fwapi.DriverTech.com/CommonTypes/2009/04">
        <Carrier DOTNumber="77949" IsDisabled="false" Id="1" CarrierName="United Van Lines LLC"
          <Address>
            <Address>1 United Drive</Address>
            <City>Fenton</City>
            <State>MO</State>
            <Zip>63026</Zip>
          </Address>
          <AuthorityGroup Name="United Van Lines" Id="1"/>
          <OperationalAuthority IsIntrastate="true" IsDisabled="false" Code="MC67234" Id="1" Alias
        </Carrier>
        <Carrier DOTNumber="125563" IsDisabled="false" Id="10" CarrierName="Mayflower Transit L
```

Example using default namespace of AdminService_v2 Namespace after 01/25/2018 v2.0.101:



```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:ns="http://fwapi.DriverTech.com/CommonTypes/2009/04"
xmlns:dat="http://fwapi.DriverTech.com/AdminService_v2">
  <soapenv:Header>
    <ns:RequestHeader>
      <ns:CompanyCode>Unigroup</ns:CompanyCode>
      [redacted]
    </ns:RequestHeader>
  </soapenv:Header>
  <soapenv:Body>
     <dat:Carriers_GetAll/>
  </soapenv:Body>
</soapenv:Envelope>
```

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/" xmlns:xsi="http://www.w
  <soap:Header>
    <ResponseHeader xmlns="http://fwapi.DriverTech.com/CommonTypes/2009/04">
      <DriverTechAPIVersion>2.0.101.22339</DriverTechAPIVersion>
      <RequestId>TEST</RequestId>
      <ExecutionMilliseconds>31.190900000000003</ExecutionMilliseconds>
    </ResponseHeader>
  </soap:Header>
  <soap:Body>
    <Carriers_GetAllResponse xmlns="http://fwapi.DriverTech.com/AdminService_v2">
      <Carriers_GetAllResult xmlns="http://fwapi.DriverTech.com/CommonTypes/2009/04">
        <Carrier DOTNumber="77949" IsDisabled="false" Id="1" CarrierName="United Van Lines LLC"
          <Address>
            <Address>1 United Drive</Address>
            <City>Fenton</City>
            <State>MO</State>
            <Zip>63026</Zip>
          </Address>
          <AuthorityGroup Name="United Van Lines" Id="1"/>
          <OperationalAuthority IsIntrastate="true" IsDisabled="false" Code="MC67234" Id="1" Alias
        </Carrier>
        <Carrier DOTNumber="125563" IsDisabled="false" Id="10" CarrierName="Mayflower Transit LI
```

- **NEW in v2.0.100 - 11/01/2017**
  - Update to support new backend framework. No API changes are being made at this time
- **NEW in v2.0.90**
  - FMCSA Mandate Pre-Release
- **NEW 02/2015**: Support for Multiple Operating Authority and Hierarchical Reporting Groups
  - Terminal Setting configurations(s)
  - Authority Group Membership management
  - Existing ReportingGroup_GetAll operation will automatically include Hierarchical Reporting Groups and Membership (if enabled). Expand the following section for detail on how the Heirarchical groupId is comprised using a HIGH WORD and LOW WORD in a single integer value. This example also includes example code for encoding/decoding the value

    - The ID of a Hierarchical reporting group will be comprised using "Hi word, Lo word": (CompanyId * 2^16) + TerminalId
      This will produce result where:
      - Value does not complete with existing Custom Reporting Groups
      - Value is a positive integer (does not require any change to Vendor software)
      - Value is easily/accurately encoded and decoded to/from HiLo value
      - CompanyId has a ceiling of 32768 (set at 30k to provide a buffer if needed in the future)
      - TerminalId has a ceiling of 65535 (set at 60k to provide a buffer if needed in the future)

        Examples:
        1. A Company Level Hierarchical Reporting Group:
           CompanyId = 201
           CompanyName = BurgerKing
           ======================================
           HierarchicalReportingGroupId = 13172736
           HierarchicalReportingGroupName = "BurgerKing" *(Company Name Only)*

        2. A Terminal Level Hierarchical Reporting Group:
           CompanyId = 201
           CompanyName = BurgerKing
           TerminalId = 54
           TerminalName = Denver102
           ======================================
           HierarchicalReportingGroupId = 13172790
           HierarchicalReportingGroupName = "BurgerKing - Denver102"

      - Existing structure is not affected:
        blocked URL

Encoding and Decoding LOW HIGH WORD value - this does not apply the additional ^ or root math

```
             namespace LOWHIGHWORD_EncodeDecodeId
{
    class Program
    {
        static void Main(string[] args)
        {
            if(args == null || args.Length < 1)
            {
                Console.WriteLine("Provide Arguments for either Encoding or Decoding as follows: ");
                Console.WriteLine("  To Encode: LOWvalue HIGHvalue(optional, defaults to 1)");
                Console.WriteLine("  To Decode: value");
            }
            else if(args[0].Length <= 4)
            {
                var encoded = MAKELONG(Convert.ToInt16(args[0].Replace("DT", "")), args.Length == 2 ?
Convert.ToInt16(args[1]) : (short)1);
                var low = LOWORD(encoded);
                var high = HIWORD(encoded);

                Console.WriteLine($"EncodedId for LOW {low} and HIGH {high} is {encoded}");
            }
            else if(args[0].Length > 4)
            {
                var encoded = Convert.ToInt32(args[0]);
                Console.WriteLine($"EncodedId {encoded} contains LOW {LOWORD(encoded)} and HIGH {HIWORD
(encoded)}");
            }
        }

        static short MAKEWORD(byte a, byte b)
        {
            return ((short)(((byte)(a & 0xff)) | ((short)((byte)(b & 0xff))) << 8));
        }

        public static byte LOBYTE(short a)
        {
            return ((byte)(a & 0xff));
        }

        public static byte HIBYTE(short a)
        {
            return ((byte)(a >> 8));
        }

        public static int MAKELONG(short a, short b)
        {
            return (((int)(a & 0xffff)) | (((int)(b & 0xffff)) << 16));
        }

        public static short HIWORD(int a)
        {
            return ((short)(a >> 16));
        }

        public static short LOWORD(int a)
        {
            return ((short)(a & 0xffff));
        }
    }
}
```

API Examples