# Working with the createXXX Sample Applications

## Introduction

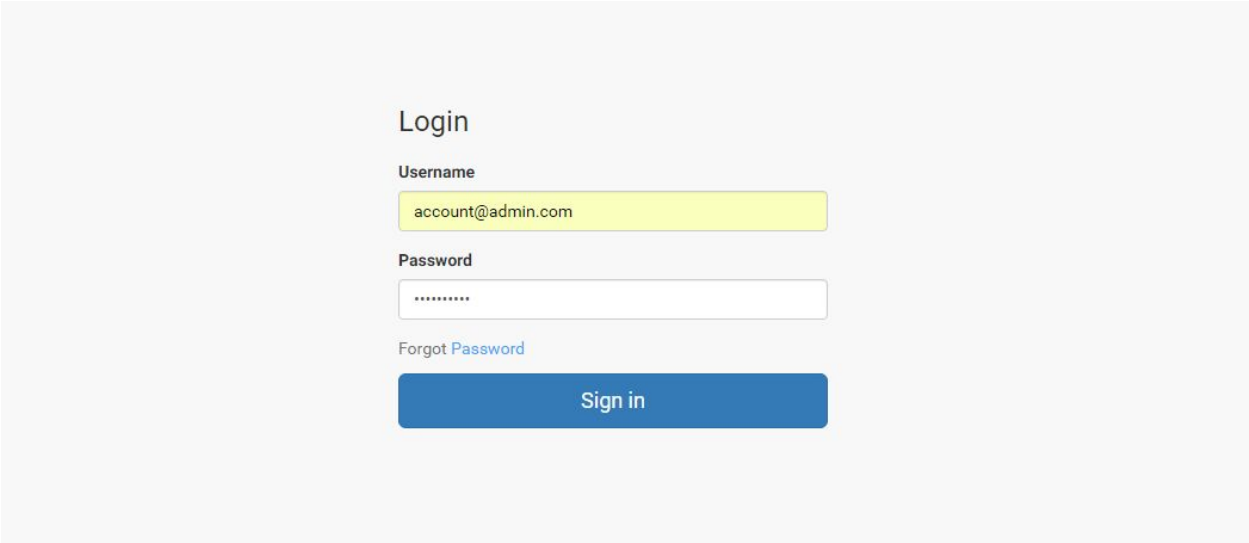
This directory provides examples that illustrate the use of the VisTracks API to create several key data structures. All examples consist of a web page (.htm) that can be launched directly from a file system. (For example, from a file explorer, you can right-click on a particular ".htm" file and open a browser.) Each web page references a pair of Javascript files that make the actual API calls and display the results.

The first Java script file is specific to the individual data object (e.g., JobSite, DriverDaily, WorkOrder, etc.) This file, (e.g., createUser.js) provides the information specific to that object. The second file contains the "core" logic that is common to all objects. It contains the code that handles the creation of the dynamic parts of the sample web page, and orchestrates the "posting" of the data to the server.

## Running a Sample Application

Before running a particular example application, you'll need to log in to the VisTracks system from a separate tab in your browser. This will establish the authentication credentials for your API session.

Assuming, for example, that you have an account on the primary VisTracks server, you would access the site at http://hos.vistracks.com which would present a login page:




You can enter your credentials which will take you to a landing page. You don't need to do anything else on the VisTracks site.

At this point, you can open a separate tab in the browser and enter the path to one of the "create*.htm" pages in this directory. For example, if you opened the "createUser.htm" page, you should see an interface similar to the illustration below.


This screen allows you to enter some of the key values for a "user" object in the system in text boxes, then issue the API command to actually create a new instance of the object. In the example below, the email address was changed slightly. You can see the JSON data that was sent to the server, and the corresponding response.

VisTracks

## Create New User

**Log into a VisTracks account in another tab within this browser, then click the create button.**

| | |
|---|---|
| Email: | johndoejr@mail.com |
| First Name: | John |
| Last Name: | Doe |
| Suffix: | Jr. |
| Password: | password |

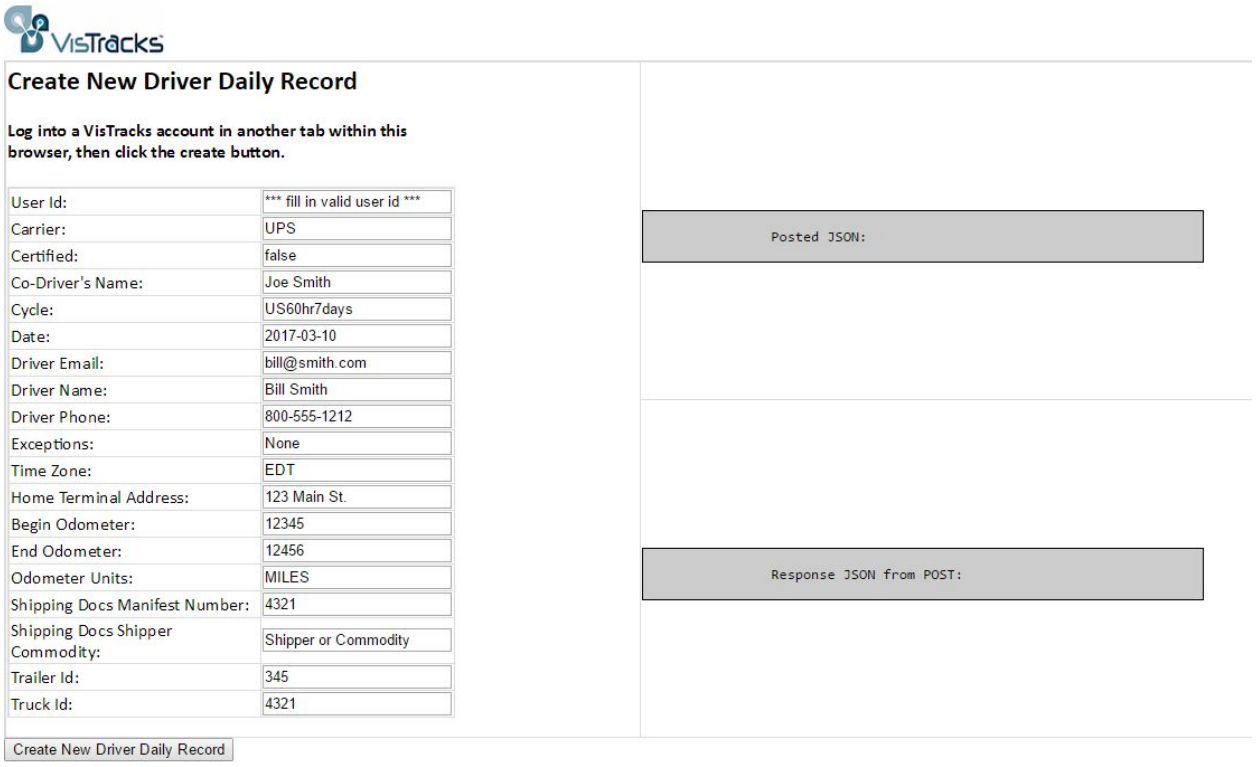
Create New User


```
Posted JSON: [
  {
    "email": "johndoejr@mail.com",
    "firstName": "John",
    "lastName": "Doe",
    "suffix": "Jr.",
    "password": "password",
    "userRoles": [
      {
        "id": 1500004
      }
    ],
    "visibilitySetIds": [
      2673
    ]
  }
]
```

```
Response JSON from POST: [
  {
    "firstName": "John",
    "lastName": "Doe",
    "suffix": "Jr.",
    "email": "johndoejr@mail.com",
    "eulaAcceptedDate": "2017-03-10T05:34:49.423Z",
    "roles": [],
    "userRoles": [
      "USER_ROLE_ASSETADMIN"
    ],
    "permissions": [
      {
        "name": "PERM_VIEW_PORTAL_UNIDENTIFIED_DRIVING_EVENTS_TAB",
        "category": "Unidentified Driving Events",
        "system": false,
        "id": 17
      },
      {
        "name": "PERM_VIEW_PORTAL_VIOLATIONS_TAB",
        "category": "Violations",
        "system": false,
        "id": 16
      },
      {
        "name": "PERM_VIEW_PORTAL_DVIR_HISTORY_TAB",
        "category": "DVIR History",
        "system": false,
        "id": 19
      },
      {
        "name": "PERM_VIEW_PORTAL_REPORTS_TAB",
        "category": "Reports",
        "system": false,
        "id": 18
      },
      {
        "name": "PERM_IS_ASSET_ADMIN",
        "category": "Users",
        "system": false,
        "id": 7
      },
      {
        "name": "PERM_IS_ELEVATED_USER",
        "category": "Users",
        "system": false,
        "id": 8
      },
      {
        "name": "PERM_VIEW_ALL_TERMINALS",
        "category": "Terminals",
        "system": false,
        "id": 11
      },
      {
        "name": "PERM_EDIT_USER_PASSWORDS",
        "category": "User Admin",
        "system": false,
        "id": 12
      },
      {
        "name": "PERM_VIEW_PORTAL_DRIVERS_TAB",
        "category": "Drivers",
        "system": false,
        "id": 13
      },
      {
        "name": "PERM_VIEW_PORTAL_DRIVER_LOGS_TAB",
        "category": "Driver Logs",
        "system": false,
        "id": 14
      },
      {
        "name": "PERM_EDIT_PORTAL_DRIVER_LOGS_TAB",
        "category": "Driver Logs",
        "system": false,
        "id": 15
      }
    ],
    "visibilitySetIds": [
      2673
    ],
    "active": true,
    "accountId": 1500001,
    "id": 1500229,
    "lastChangedDate": "2017-03-10T05:34:49.425Z"
  }
]
```

Note in particular that among the values returned in the response, there is one element labeled “id”. In this example, the returned “id” has the value “1500229”. This represents an internal identifier for that object and will be used in some cases to establish a link between two structures.

For example, if we look at the “Driver Daily Record”, another type of object in the system, we’ll see that one of the fields required to create an instance of this record is a reference to a User Id.



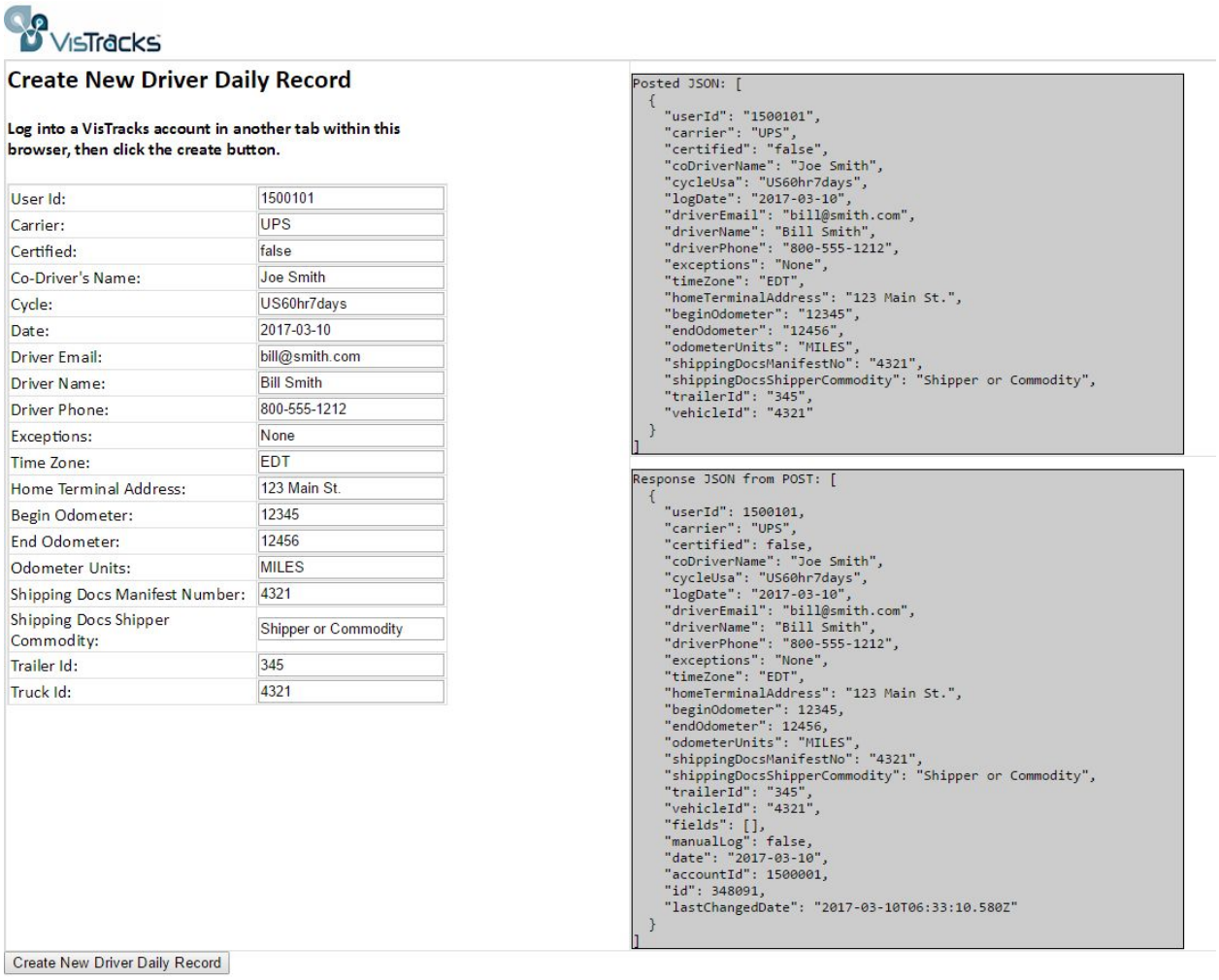

In order to successfully create an instance of the “Driver Daily” record, we would need to associate a valid id of a user record. In this case, we could fill in the value from above:

| | |
|---|---|
| User Id: | 1500229 |
| Carrier: | UPS |
| Certified: | false |
| Co-Driver's Name: | Joe Smith |
| Cycle: | US60hr7days |
| Date: | 2017-03-10 |

Submitting this record will now create a new record on the server linking the “Driver Daily” report to the given user.

## Summary

The VisTracks API is based on the RESTful approach to accessing remote web services. While it is possible to retrieve information from the server using an HTTP “GET”, creating new objects requires that a properly formed JSON structure be posted to the server. These examples provide a starting point to explore the basic fields that are required to successfully create the key objects that form the basis of the system.




As before, if you examine the returned JSON, you'll see that this object has also been assigned a unique identifier 348091 by the server.