Swagger UI

Last Modified on 11/03/2026 9:29 am EST

1place Swagger UI - How to Test API Integrations

Purpose

Swagger UI is an interactive tool used to view and test **1place API endpoints**. It allows users to submit API requests and review the responses returned by 1place.

This tool can be used to test API communication and verify whether a particular 1place integration endpoint is responding as expected.

Available 1place Integrations

The following 1place integrations can be tested using Swagger UI:

- CCC
- OPSTrax
- PartsTraderRetail
- PartsTraders

How to Test a 1place API Endpoint Using Swagger UI

1. **Locate the API Endpoint**

Open the 1place Swagger UI and locate the integration and API endpoint you want to test.

Expand the endpoint to view its available parameters and testing options.

2. **Enter the Password**

   Enter the password normally used for the 1place API request in the appropriate request field, if required by the endpoint.

3. **Populate the Request Parameters**

   Review the sample data displayed for the request parameters.

   Click the sample data type or value (such as `string`, `0`, or `true`). Swagger UI will automatically populate the request body with the sample structure.

   For example:

```
{
 "listofitems": [
  "string"
 ],
 "CompanyID": 0,
 "ThirdPartyType": "string",
 "ThirdPartyCustomerID": "string",
 "Zipcode": "string",
 "Quantity": 0,
 "RequireValidCustomer": true
}
```

   Replace the sample values with the actual information needed for your test.

   Common sample values include:

   - `string` – Text value.
   - `0` – Numeric value.
   - `true` – Boolean value (True or False).
   - `listofitems` – A list or array of values.

4. **Click Try it out!**

   Once the required information has been entered, click **Try it out!** to submit the API request.

5. **Review the API Response**

Swagger UI will display the result of the API request.

Review the response status and returned data to verify whether the 1place endpoint responded successfully and returned the expected information.

If an error is returned, review the response details for additional information.

Important Notes

- Make sure you select the correct integration and API endpoint before submitting a request.
- Use the appropriate parameter values and data types required by the selected endpoint.
- Sample values such as `string` and `0` are placeholders. Replace them with actual test information.
- A successful API response confirms that the tested request received a successful response. It does not necessarily mean the entire integration workflow is functioning correctly.
- Treat API passwords and other credentials as sensitive information. Do not include actual passwords in screenshots, shared examples, or KBA documentation.