

LC360 Carrier API V1.00

The Loss Control 360 Carrier API utilizes REST services to communicate on the API. The available services are described at the API Help Url provided to you.

<https://YourLC360Vendor/API/Carrier/V1/Help>

This help page will describe the different methods available to you via the API web service. Each method will require the credentials of the Customer Manager account for LC360 to be provided with each request. All methods utilize the post method and allow either XML or JSON to be provided as the body of the request.

NOTE: All services will throw an error code 400 BadRequest if they are missing you credentials or have a malformed request.

The web service has documented proxy classes available for reference at:

<https://YourLC360Vendor/API/Carrier/V1/Reference/>

Available Methods

ImportInspections

<https://YourLC360Vendor/API/Carrier/V1/help/operations/ImportInspections>

The ImportInspections method gives the carrier the ability to directly push inspection work to the vendor. You may use JSON or XML to create the request. The below elements marked in red are required. At least one of telephone numbers should be provided.

Depending on the vendors setup of your account you may need to include Agent, some Attributes, or some Extra Fields if defined.

Example JSON Format

```
{
  "Password":"String content",
  "UserName":"String content",
  "Inspections":{
    "Agent":{
      "Address":{
        "City":"String content",
        "StateOrProvince":"String content",
        "Street1":"String content",
        "Street2":"String content",
        "ZipCode":"String content"
      }
    }
  }
}
```

```
    },
    "AgencyName":"String content",
    "AgentCode":"String content",
    "ContactName":"String content",
    "Fax":"String content",
    "Phone":"String content"
  },
  "Attributes":{
    "BuildingCost":12678967,
    "BusinessTotalRevenue":12678967,
    "BusinessType":"String content",
    "ContentsCost":12678967.543233,
    "CoverageAln":12678967.543233,
    "IsoClass":"String content",
    "Occupancy":"String content",
    "YearBuilt":2147483647
  },
  "CustomerKey":"String content",
  "EffectiveDate":"\\Date(928164000000-0400)\\",
  "ExtraFields":[{
    "Key":"String content",
    "Value":"String content"
  }],
  "IgnoreDuplicates":true,
  "InspectionType":"String content",
  "IsRush":true,
  "Location":{
    "City":"String content",
    "StateOrProvince":"String content",
    "Street1":"String content",
    "Street2":"String content",
    "ZipCode":"String content"
  },
  "Mailing":{
    "City":"String content",
    "StateOrProvince":"String content",
    "Street1":"String content",
    "Street2":"String content",
    "ZipCode":"String content"
  },
  "Notes":"String content",
  "PolicyHolder":{
```

```

        "CellPhone":"String content",
        "HomePhone":"String content",
        "PolicyHolderContact":"String content",
        "PolicyHolderName":"String content",
        "WorkPhone":"String content"
    },
    "PolicyNumber":"String content",
    "SupplementForms":["String content"],
    "Underwriter":{
        "CorrespondanceEmail":"String content",
        "FirstName":"String content",
        "IgnoreDuplicate":true,
        "LastName":"String content",
        "Phone":"String content",
        "ReportEmail":"String content"
    }
}
}

```

Example JSON Response

```

[[
    {
        "CaseID":"1627aea5-8e0a-4371-9022-9b504344e724",
        "CaseNumber":2147483647,
        "Duplicate":true,
        "DuplicateReason":{
            "AddressSame":true,
            "DaysOld":2147483647,
            "PolicyNumberSame":true
        },
        "Errors":[{
            "ErrorText":"String content",
            "ErrorType":0
        }],
        "PolicyNumber":"String content",
        "Successful":true
    }
]]

```

Each response object in the array must be reviewed to make sure each item successfully imported. If the Duplicate is true you can send the inspection in with *IgnoreDuplicates = true* to have the inspection inserted ignoring the fact there is currently an open item that matches the policy number or address.

Any errors will need to be reviewed and fixed before the inspection will import successfully.

GetUpdate

Will provide you with an updated status for the queried inspection as well as any new notes made available by the vendor for that inspection.

CompletedInspections

Will return a list of inspection that match the CustomerKeys provided (or all if none provided) that have not yet been marked exported via the MarkCompletedInspectionsExortedRequest method.

CompletedInspectionPhotos

Will provide all photos included by the inspector of the home.

CompletedInspectionPDF

Will return the final PDF report for the completed inspection.

CompletedInspectionFormMeta

Will return the available form data from the completed forms on the inspection. Note not all form data is available in this format (RCT, BVC etc)

CompletedInspectionFormDataSet

Returns an XML serialized DataSet of the available form data. Each form will have a representative Table if data is available for the form.

MarkCompletedInspectionsExortedRequest

Will mark an inspection exported so that it will not show up in subsequent calls to CompletedInspections.