Laboratorio - Integrar una API REST en una aplicación Python

Objetivos

Parte 1: Iniciar la máquina virtual de DEVASC.

Parte 2: Demostrar la Aplicación de Direcciones de MapQuest

Parte 3: Obtener una clave API de MapQuest

Parte 4: Crear la aplicación de Dirección Básica de MapQuest

Parte 5: Actualizar la Aplicación de Dirección de MapQuest con más características

Parte 6: Probar la funcionalidad completa de la aplicación

1. Aspectos básicos/Situación

En este laboratorio, creará una aplicación en Visual Studio Code (VS Code) que recupera datos JSON de la API de Direcciones de MapQuest, analiza los datos y les da formato para la salida al usuario. Utilizará la solicitud GET Route de la API de direcciones de MapQuest. Revise la documentación de GET Route Directions API aquí:

<https://developer.mapquest.com/documentation/directions-api/route/get/>

**Nota**: Si el enlace anterior ya no funciona, busque por “Documentación de la API de MapQuest”.

1. Recursos necesarios

* Una computadora con el sistema operativo de su elección.
* VirtualBox o VMware.
* Máquina virtual (Virtual Machine) DEVASC.

# Instrucciones

## Inicie la máquina virtual de DEVASC.

Inicie la máquina virtual DEVASC, en este laboratorio usará Visual Studio para crear el script en Python, luego tendrá que publicar el script en su cuenta de GitHub.

* **Modifique todas las variables de ejemplo que se muestran a lo largo de esta guía de laboratorio, excepto cuando solicite datos hacia la API de MapQuest.**
* **Los textos de ejemplo “Print” deben estar en español**
* **Las variables deben indicar por ejemplo:**
  + **Main\_api:** api\_nombrealumno
  + **Orig:** origen\_nombrealumno
  + **Dest:** destino\_nombrealumno
  + **Key:** llave\_nombrealumno

**El Script Final debe ser publicado en su cuenta de Github**

**Este documento WORD debe subirlo al AAI en el recurso OMRA05\_EVA02\_PARTE2\_ENTREGA**

## Demostrar la Aplicación de Direcciones de MapQuest

En esta actividad creará este script paso a paso en este laboratorio, la aplicación solicitará una ubicación inicial y un destino. A continuación, solicitará datos JSON de la API de Direcciones de MapQuest, los analiza y muestra información útil.

Esta es la información de salida del script que creará:

|  |
| --- |
| Lugar de inicio: **Washington**  Destino: **Baltimore**  URL: https://www.mapquestapi.com/directions/v2/route?key =your\_api\_key&from=Washington&to=Baltimore  Estado de la API: 0 = Una llamada de ruta exitosa.  Cómo llegar desde Washington a Baltimore  Duración del viaje: 00:49:19  Kilómetros: 61.32  Combustible utilizado (Litros): 6.24  =============================================  Salga hacia el norte por la 6ª ST/US-50 E/US-1 N hacia Pennsylvania Ave/US-1 Alt N. (1,28 km)  Gire a la derecha en New York Ave/US-50 E. Y continúe por US-50 E (cruce hacia Maryland). (7.51 km)  Tome la salida Balt-Wash Parkway a la izquierda hacia Baltimore. (0.88 km)  Incorpórese en MD-295 N. (50.38 km)  Gire a la derecha hacia W Pratt St. (0,86 km)  Gire a la izquierda hacia S Calvert ST/MD-2. (0.43 km)  Bienvenido a BALTIMORE, MD. (0.00 km)  =============================================  Ubicación inicial: salir |

**Nota**: Para ver el JSON para la salida anterior, puede copiar la URL en una pestaña del navegador. Sin embargo, deberá reemplazar **your\_api\_key por** la clave API de MapQuest que obtenga en la Parte 3.

## Obtener una clave de API de MapQuest

Antes de crear la aplicación, debe completar los siguientes pasos para obtener una clave de API de MapQuest.

* + - 1. Vaya a: <https://developer.mapquest.com/>.
      2. Haga clic en **Registrarse** arriba de la página
      3. Rellene el formulario para crear una nueva cuenta. Para **compañía**, ingrese **Estudiante de Cisco Networking Academy**.
      4. Después de hacer clic en **Registrarme**, se le redirigirá a la página **Gestionar claves**. Si se le redirige a otro lugar, haga clic en **Manage Keys** en la lista de opciones de la izquierda.
      5. Haga clic en **Approve All Keys**.
      6. Expanda **My Application**.
      7. Copie su **clave de consumidor** en un archivo de texto para su uso futuro. Esta será la clave que usted usará para el resto de este laboratorio.

**Nota**: MapQuest puede cambiar el proceso para obtener una clave. Si los pasos anteriores ya no son válidos, busque en Internet “pasos para generar el clave api de mapquest”.

## Crear la aplicación de dirección básica de MapQuest

En esta parte, creará una secuencia de comandos Python para enviar una solicitud de URL a la API de direcciones de MapQuest. A continuación, va a probar su llamada a la API. A lo largo del resto de este laboratorio, creará el script en partes, guardando el archivo con un nombre nuevo cada vez. Esto ayudará con el aprendizaje de las piezas de la aplicación, así como le proporcionará una serie de scripts a los que puede volver si tiene algún problema en la versión actual de su aplicación.

### Cree un nuevo archivo en Visual Studio Code (VS Code).

Puede usar cualquier herramienta que desee introducir en los comandos Python y ejecutar el código Python. Sin embargo, este laboratorio demostrará la creación de la aplicación en VS Code.

* + - 1. Abrir **VS Code.** Hay un acceso directo en el **escritorio**, para su comodidad.
      2. Seleccione Archivo > Abrir carpeta...
      3. Vaya al directorio **~/labs/devnet-src/mapquest** y haga clic en **Aceptar**. Este directorio está actualmente vacío y es donde almacenará cada iteración de su aplicación.
      4. Seleccionar **File** > **New File**.
      5. Seleccionar **File Guardar como...,** nombrar el archivo **mapquest\_parse-json\_1.py** y haga clic en **Save**.

### Importación de módulos para la aplicación.

Para comenzar su script para analizar datos JSON, necesitará importar dos módulos de la biblioteca de Python: **requests** y **urllib.parse**. El módulo de **requests** proporciona funciones para recuperar datos JSON de una URL. El módulo **urllib.parse** proporciona una variedad de funciones que le permitirán analizar y manipular los datos JSON que recibe de una solicitud a una URL.

* + - 1. Agregue las siguientes instrucciones de importación en la parte superior del script.

import urllib.parse

import requests

* + - 1. Seleccione **Terminal > New Terminal** para abrir una Terminal dentro del VS Code.
      2. Guardar y ejectuar el script. No debería recibir errores. Debe guardar y ejecutar los scripts a menudo para probar la funcionalidad del código.

devasc @labvm: ~/labs/devnet-src/mapquest$ **python3 mapquest\_parse-json\_1.py** devasc @labvm: ~/labs/devnet-src/mapquest$

### Crea la URL de la solicitud a la API de direcciones de MapQuest.

El primer paso para crear su solicitud de API es construir la URL que su aplicación utilizará para realizar la llamada. Inicialmente, la URL será la combinación de las siguientes variables:

* + **main\_api** - la URL principal a la que está accediendo
  + **orig** - el parámetro para especificar su punto de origen
  + **dest** - el parámetro para especificar su destino
  + **key** : la clave API de MapQuest que ha recuperado del sitio web del desarrollador
    - 1. Cree variables para generar la URL que se enviará en la solicitud. En el código siguiente, reemplace **your\_api\_key por la clave** de consumidor que copió de su cuenta de desarrollador de MapQuest.

main\_api = “https://www.mapquestapi.com/directions/v2/route?”

orig = “Washington, D.C.”

dest = “Baltimore, Md”

clave = “tu\_api\_key”

* + - 1. Combine las cuatro variables **main\_api**, **orig**, **dest**y **key** para dar formato a la URL solicitada. Utilice el método **urlencode** para formatear correctamente el valor de la dirección. Esta función construye la parte de parámetros de la URL y convierte posibles caracteres especiales en el valor de la dirección en caracteres aceptables (por ejemplo, espacio en “+” y una coma en “%2C”).

url = main\_api + urllib.parse.urlencode ({“key” :key, “from” :orig, “to” :dest})

* + - 1. Crear una variable para contener la respuesta de la URL solicitada e imprimir los datos JSON devueltos. La variable **json\_data** contiene una representación del diccionario de Python de la respuesta **json** del método **get** del módulo de **peticiones**. El **requests.get** realizará la llamada API a la API de MapQuest. La declaración de **print** se utilizará temporalmente para comprobar los datos devueltos. Reemplazará esta declaración de impresión con opciones de visualización más sofisticadas más adelante en el laboratorio.

json\_data = requests.get (url) .json ()

print (json\_data)

* + - 1. Agregue su código final hasta el momento, destaque su clave personal.

|  |
| --- |
| import urllib.parse  import requests  main\_api\_benjamincornejo = "https://www.mapquestapi.com/directions/v2/route?"  orig\_benjamincornejo = "Washington, D.C."  dest\_benjamincornejo = "Baltimore, Md"  key\_benjamincornejo = "NBRA305IVkuxUEwkz8zoXjBN6SeNyqPr"  url = main\_api\_benjamincornejo + urllib.parse.urlencode ({"key" :key\_benjamincornejo , "from" :orig\_benjamincornejo , "to" :dest\_benjamincornejo})  json\_data = requests.get (url) .json ()  print (json\_data) |

### Pruebe la solicitud de URL.

* + - 1. Guarde y ejecute su script **mapquest\_parse-json\_1.py** y verifique que funcione.
      2. Solucione el problema de su código, si es necesario. Aunque su salida puede ser ligeramente diferente, debería obtener una respuesta JSON similar a la siguiente. Observe que la salida es un diccionario con dos pares clave/valor. El valor de la clave **route** es otro diccionario que incluye diccionarios y listas adicionales. La clave **información** incluye el par de clave/valor de **código de estado que usará** más adelante en el laboratorio.

devasc @labvm: ~/labs/devnet-src/mapquest$ **python3 mapquest\_parse-json\_1.py**

{'route': {'hasTollRoad': False, 'hasBridge': True, 'boundingBox': {'lr': {'lng': -76.612137, 'lat': 38.892063}, 'ul': {'lng': -77.019913, 'lat': 39.290443}}, 'distance': 38.089, 'hasTimedRestriction': False, 'hasTunnel': False, 'hasHighway': True, 'computedWaypoints': [], 'routeError': {'errorCode': -400, 'message': ''}, 'formattedTime': '00:49:29', 'sessionId': '5eadfc17-00ee-5f21-02b4-1a24-0647e6e69816', 'hasAccessRestriction': False, 'realTime': 2915, 'hasSeasonalClosure': False, 'hasCountryCross': False, 'fuelUsed': 1.65, 'legs': [{'hasTollRoad': False, 'hasBridge': True, 'destNarrative': 'Proceed to BALTIMORE, MD.', 'distance': 38.089, 'hasTimedRestriction': False, 'hasTunnel': False, 'hasHighway': True, 'index': 0, 'formattedTime': '00:49:29', 'origIndex': -1, 'hasAccessRestriction': False, 'hasSeasonalClosure': False, 'hasCountryCross': False, 'roadGradeStrategy': [[]], 'destIndex': 3, 'time': 2969, 'hasUnpaved': False, 'origNarrative': '', 'maneuvers': [{'distance': 0.792, 'streets': ['6th St', 'US-50 E', 'US-1 N'], 'narrative': 'Start out going north on 6th St/US-50 E/US-1 N toward Pennsylvania Ave/US-1 Alt N.', 'turnType': 0, 'startPoint': {'lng': -77.019913, 'lat': 38.892063}, 'index': 0, 'formattedTime': '00:02:06', 'directionName': 'North', 'maneuverNotes': [], 'linkIds': [], 'signs': [{'extraText': '', 'text': '50', 'type': 2, 'url': 'http://icons.mqcdn.com/icons/rs2.png?n=50&d=EAST', 'direction': 8}, {'extraText': '', 'text': '1', 'type': 2, 'url': 'http://icons.mqcdn.com/icons/rs2.png?n=1&d=NORTH',

<<<<< >>>>>

output omitted

<<<<< >>>>>

'geocodeQuality': 'CITY', 'adminArea1Type': 'Country', 'adminArea3Type': 'State', 'latLng': {'lng': -76.61233, 'lat': 39.29044}}], 'time': 2969, 'hasUnpaved': False, 'locationSequence': [0, 1], 'hasFerry': False}, 'info': {'statuscode': 0, 'copyright': {'imageAltText': '© 2019 MapQuest, Inc.', 'imageUrl': 'http://api.mqcdn.com/res/mqlogo.gif', 'text': '© 2019 MapQuest, Inc.'}, 'messages': []}}

devasc@labvm:~/labs/devnet-src/mapquest$

* + - 1. Cambie las variables **origen** y **destino** a direcciones de Chile y la quinta región. Vuelva a ejecutar el script para obtener resultados diferentes. Para garantizar los resultados que desea, lo mejor es incluir tanto la ciudad como el estado para las ciudades de los Estados Unidos. Al referirse a ciudades de otros países, normalmente puede usar el nombre en inglés de la ciudad y el país o el nombre nativo. Por ejemplo:

orig = "Rome, Italy"

dest = "Frascati, Italy"

or

orig = "Roma, Italia"

dest = "Frascati, Italia"

Adjunte evidencia de la consulta realizada dentro de Chile y otra consulta dentro de la Quinta región.

|  |
| --- |
| Consulta ciudades en Chile |
| orig\_benjamincornejo = "concepcion, chile"  dest\_benjamincornejo = "la serena, chile"  devasc@labvm:~/labs/devnet-src/mapquest$ python3 /home/devasc/labs/devnet-src/mapquest/mapquest\_parse-json\_1.py   * + - 1. {'route': {'sessionId': 'AI8A5wcAAH4AAAAAAAAAEgAAAAIBAAB42mNgYmRgZGJgYGDPSC1KtUrOXTu9XhTIZeA2VW\_kCsnn9bmw-LxqKpDeCqQZsACYxtSOXLBGKeGXjUzd0qG-8\_P3aIHoeUCaAQdYn-G46S0zkJHXL6C4xomBvZVHgIODgWqgAbuwIAMbgyMLhN2cxAAAsSggRVidKCk:car', 'realTime': 36372, 'distance': 603.5285, 'time': 33936, 'formattedTime': '09:25:36', 'hasHighway': True, 'hasTollRoad': True, 'hasBridge': True, 'hasSeasonalClosure': False, 'hasTunnel': True, 'hasFerry': False, 'hasUnpaved': False, 'hasTimedRestriction': False, 'hasCountryCross': False, 'legs': [{'index': 0, 'hasTollRoad': True, 'hasHighway': True, 'hasBridge': True, 'hasUnpaved': False, 'hasTunnel': True, 'hasSeasonalClosure': False, 'hasFerry': False, 'hasCountryCross': False, 'hasTimedRestriction': False, 'distance': 603.5285, 'time': 36372, 'formattedTime': '10:06:12', 'origIndex': 0, 'origNarrative': '', 'destIndex': 0, 'destNarrative': '', 'maneuvers': [{'index': 0, 'distance': 0.1901, 'narrative': "Head toward Calle Rengo on Avenida Bernardo O'Higgins. Go for 0.2 mi.", 'time': 82, 'direction': 7, 'directionName': 'West', 'signs': [], 'maneuverNotes': [], 'formattedTime': '00:01:22', 'transportMode': 'car', 'startPoint': {'lat': -36.828152, 'lng': -73.051199}, 'turnType': 0, 'attributes': 0, 'iconUrl': '', 'streets': ["Avenida Bernardo O'Higgins"], 'mapUrl': 'https://www.mapquestapi.com/staticmap/v5/map?key=NBRA305IVkuxUEwkz8zoXjBN6SeNyqPr&size=225,160&locations=-36.828152,-73.051199|marker-1||-36.82943000000001,-73.05429|marker-2||&center=-36.82879100000001,-73.05274449999999&defaultMarker=none&zoom=15&session=AI8A5wcAAH4AAAAAAAAAEgAAAAIBAAB42mNgYmRgZGJgYGDPSC1KtUrOXTu9XhTIZeA2VW\_kCsnn9bmw-LxqKpDeCqQZsACYxtSOXLBGKeGXjUzd0qG-8\_P3aIHoeUCaAQdYn-G46S0zkJHXL6C4xomBvZVHgIODgWqgAbuwIAMbgyMLhN2cxAAAsSggRVidKCk:car'}, {'index': 1, 'distance': 0.0764, 'narrative': "Continue on Calle Bernardo O'Higgins. Go for 404 ft.", 'time': 22, 'direction': 6, 'directionName': 'Southwest', 'signs': [], 'maneuverNotes': [], 'formattedTime': '00:00:22', 'transportMode': 'car', 'startPoint': {'lat': -36.82943000000001, 'lng': -73.05429}, 'turnType': 0, 'attributes': 0, 'iconUrl': '', 'streets': ["Calle Bernardo O'Higgins"], 'mapUrl': 'https://www.mapquestapi.com/staticmap/v5/map?key=NBRA305IVkuxUEwkz8zoXjBN6SeNyqPr&size=225,160&locations=-36.82943000000001,-73.05429|marker-2||-36.82996000000001,-73.05551999999999|marker-3||&center=-36.82969500000001,-73.05490499999999&defaultMarker=none&zoom=16&session=AI8A5wcAAH4AAAAAAAAAEgAAAAIBAAB42mNgYmRgZGJgYGDPSC1KtUrOXTu9XhTIZeA2VW\_kCsnn9bmw-LxqKpDeCqQZsACYxtSOXLBGKeGXjUzd0qG-8\_P3aIHoeUCaAQdYn-G46S0zkJHXL6C4xomBvZVHgIODgWqgAbuwIAMbgyMLhN2cxAAAsSggRVidKCk:car'}, {'index': 2, 'distance': 0.3094, 'narrative': 'Turn right onto Calle Obispo Hipólito Salas. Go for 0.3 mi.', 'time': 72, 'direction': 2, 'directionName': 'Northwest', 'signs': [], 'maneuverNotes': [], 'formattedTime': '00:01:12', 'transportMode': 'car', 'startPoint': {'lat': -36.82996000000001, 'lng': -73.05551999999999}, 'turnType': 2, 'attributes': 0, 'iconUrl': '', 'streets': ['Calle Obispo Hipólito Salas'], 'mapUrl': 'https://www.mapquestapi.com/staticmap/v5/map?key=NBRA305IVkuxUEwkz8zoXjBN6SeNyqPr&size=225,160&locations=-36.82996000000001,-73.05551999999999|marker-3||-36.825950000000006,-73.05805|marker-4||&center=-36.827955,-73.05678499999999&defaultMarker=none&zoom=14&session=AI8A5wcAAH4AAAAAAAAAEgAAAAIBAAB42mNgYmRgZGJgYGDPSC1KtUrOXTu9XhTIZeA2VW\_kCsnn9bmw-LxqKpDeCqQZsACYxtSOXLBGKeGXjUzd0qG-8\_P3aIHoeUCaAQdYn-G46S0zkJHXL6C4xomBvZVHgIODgWqgAbuwIAMbgyMLhN2cxAAAsSggRVidKCk:car'}, {'index': 3, 'distance': 0.1746, 'narrative': 'Turn left onto Avenida Los Carrera. Go for 0.2 mi.', 'time': 63, 'direction': 7, 'directionName': 'West', 'signs': [], 'maneuverNotes': [], 'formattedTime': '00:01:03', 'transportMode': 'car', 'startPoint': {'lat': -36.825950000000006, 'lng': -73.05805}, 'turnType': 6, 'attributes': 0, 'iconUrl': '', 'streets': ['Calle Obispo Hipólito Salas', 'Avenida Los Carrera'], 'mapUrl': 'https://www.mapquestapi.com/staticmap/v5/map?key=NBRA305IVkuxUEwkz8zoXjBN6SeNyqPr&size=225,160&locations=-36.825950000000006,-73.05805|marker-4||-36.82692,-73.06078|marker-5||&center=-36.826435000000004,-73.059415&defaultMarker=none&zoom=15&session=AI8A5wcAAH4AAAAAAAAAEgAAAAIBAAB42mNgYmRgZGJgYGDPSC1KtUrOXTu9XhTIZeA2VW\_kCsnn9bmw-LxqKpDeCqQZsACYxtSOXLBGKeGXjUzd0qG-8\_P3aIHoeUCaAQdYn-G46S0zkJHXL6C4xomBvZVHgIODgWqgAbuwIAMbgyMLhN2cxAAAsSggRVidKCk:car'}, {'index': 4, 'distance': 0.3654, 'narrative': 'Turn right onto Avenida Arturo Prat toward Talcahuano. Go for 0.4 mi.', 'time': 69, 'direction': 2, 'directionName': 'Northwest', 'signs': [], 'maneuverNotes': [], 'formattedTime': '00:01:09', 'transportMode': 'car', 'startPoint': {'lat': -36.82692, 'lng': -73.06078}, 'turnType': 2, 'attributes': 0, 'iconUrl': '', 'streets': ['Avenida Arturo Prat'], 'mapUrl': 'https://www.mapquestapi.com/staticmap/v5/map?key=NBRA305IVkuxUEwkz8zoXjBN6SeNyqPr&size=225,160&locations=-36.82692,-73.06078|marker-5||-36.82217000000001,-73.06372000000002|marker-6||&center=-36.824545,-73.06225&defaultMarker=none&zoom=14&session=AI8A5wcAAH4AAAAAAAAAEgAAAAIBAAB42mNgYmRgZGJgYGDPSC1KtUrOXTu9XhTIZeA2VW\_kCsnn9bmw-LxqKpDeCqQZsACYxtSOXLBGKeGXjUzd0qG-8\_P3aIHoeUCaAQdYn-G46S0zkJHXL6C4xomBvZVHgIODgWqgAbuwIAMbgyMLhN2cxAAAsSggRVidKCk:car'}, {'index': 5, 'distance': 1.3602, 'narrative': 'Continue on Avenida 21 de Mayo. Go for 1.4 mi.', 'time': 180, 'direction': 2, 'directionName': 'Northwest', 'signs': [], 'maneuverNotes': [], 'formattedTime': '00:03:00', 'transportMode': 'car', 'startPoint': {'lat': -36.82217000000001, 'lng': -73.06372000000002}, 'turnType': 0, 'attributes': 0, 'iconUrl': '', 'streets': ['Avenida 21 de Mayo'], 'mapUrl': 'https://www.mapquestapi.com/staticmap/v5/map?key=NBRA305IVkuxUEwkz8zoXjBN6SeNyqPr&size=225,160&locations=-36.82217000000001,-73.06372000000002|marker-6||-36.807400000000015,-73.07842000000004|marker-7||&center=-36.814785000000015,-73.07107000000002&defaultMarker=none&zoom=12&session=AI8A5wcAAH4AAAAAAAAAEgAAAAIBAAB42mNgYmRgZGJgYGDPSC1KtUrOXTu9XhTIZeA2VW\_kCsnn9bmw-LxqKpDeCqQZsACYxtSOXLBGKeGXjUzd0qG-8\_P3aIHoeUCaAQdYn-G46S0zkJHXL6C4xomBvZVHgIODgWqgAbuwIAMbgyMLhN2cxAAAsSggRVidKCk:car'}, {'index': 6, 'distance': 2.2432, 'narrative': 'Take ramp onto Avenida Jorge Alessandri toward Aeropuerto/Autopista Los Conquistadores. Go for 2.2 mi.', 'time': 262, 'direction': 3, 'directionName': 'Northeast', 'signs': [], 'maneuverNotes': [], 'formattedTime': '00:04:22', 'transportMode': 'car', 'startPoint': {'lat': -36.807400000000015, 'lng': -73.07842000000004}, 'turnType': 12, 'attributes': 0, 'iconUrl': '', 'streets': ['Avenida Jorge Alessandri'], 'mapUrl': 'https://www.mapquestapi.com/staticmap/v5/map?key=NBRA305IVkuxUEwkz8zoXjBN6SeNyqPr&size=225,160&locations=-36.807400000000015,-73.07842000000004|marker-7||-36.78014,-73.05810000000002|marker-8||&center=-36.79377000000001,-73.06826000000004&defaultMarker=none&zoom=11&session=AI8A5wcAAH4AAAAAAAAAEgAAAAIBAAB42mNgYmRgZGJgYGDPSC1KtUrOXTu9XhTIZeA2VW\_kCsnn9bmw-LxqKpDeCqQZsACYxtSOXLBGKeGXjUzd0qG-8\_P3aIHoeUCaAQdYn-G46S0zkJHXL6C4xomBvZVHgIODgWqgAbuwIAMbgyMLhN2cxAAAsSggRVidKCk:car'}, {'index': 7, 'distance': 2.0841, 'narrative': 'Take the 2nd exit from roundabout onto Avenida Jorge Alessandri. Go for 2.1 mi.', 'time': 191, 'direction': 3, 'directionName': 'Northeast', 'signs': [], 'maneuverNotes': [], 'formattedTime': '00:03:11', 'transportMode': 'car', 'startPoint': {'lat': -36.78014, 'lng': -73.05810000000002}, 'turnType': 28, 'attributes': 0, 'iconUrl': '', 'streets': ['Avenida Jorge Alessandri'], 'mapUrl': 'https://www.mapquestapi.com/staticmap/v5/map?key=NBRA305IVkuxUEwkz8zoXjBN6SeNyqPr&size=225,160&locations=-36.78014,-73.05810000000002|marker-8||-36.75336000000001,-73.04371|marker-9||&center=-36.76675,-73.05090500000001&defaultMarker=none&zoom=11&session=AI8A5wcAAH4AAAAAAAAAEgAAAAIBAAB42mNgYmRgZGJgYGDPSC1KtUrOXTu9XhTIZeA2VW\_kCsnn9bmw-LxqKpDeCqQZsACYxtSOXLBGKeGXjUzd0qG-8\_P3aIHoeUCaAQdYn-G46S0zkJHXL6C4xomBvZVHgIODgWqgAbuwIAMbgyMLhN2cxAAAsSggRVidKCk:car'}, {'index': 8, 'distance': 35.9942, 'narrative': 'Take ramp onto Acceso Norte A Concepción (152) toward Penco/RUTA 5. Go for 36.0 mi.', 'time': 2565, 'direction': 8, 'directionName': 'East', 'signs': [], 'maneuverNotes': [], 'formattedTime': '00:42:45', 'transportMode': 'car', 'startPoint': {'lat': -36.75336000000001, 'lng': -73.04371}, 'turnType': 12, 'attributes': 0, 'iconUrl': '', 'streets': ['152', 'Acceso Norte A Concepción', 'Autopista del Itata', 'Puente Las Ballenas', 'Autopista Interportuaria'], 'mapUrl': 'https://www.mapquestapi.com/staticmap/v5/map?key=NBRA305IVkuxUEwkz8zoXjBN6SeNyqPr&size=225,160&locations=-36.75336000000001,-73.04371|marker-9||-36.657710000000144,-72.45204000000001|marker-10||&center=-36.705535000000076,-72.74787500000001&defaultMarker=none&zoom=7&session=AI8A5wcAAH4AAAAAAAAAEgAAAAIBAAB42mNgYmRgZGJgYGDPSC1KtUrOXTu9XhTIZeA2VW\_kCsnn9bmw-LxqKpDeCqQZsACYxtSOXLBGKeGXjUzd0qG-8\_P3aIHoeUCaAQdYn-G46S0zkJHXL6C4xomBvZVHgIODgWqgAbuwIAMbgyMLhN2cxAAAsSggRVidKCk:car'}, {'index': 9, 'distance': 0.1404, 'narrative': 'Continue on Puente Itata. Go for 0.1 mi.', 'time': 9, 'direction': 8, 'directionName': 'East', 'signs': [], 'maneuverNotes': [], 'formattedTime': '00:00:09', 'transportMode': 'car', 'startPoint': {'lat': -36.657710000000144, 'lng': -72.45204000000001}, 'turnType': 0, 'attributes': 0, 'iconUrl': '', 'streets': ['Puente Itata', 'Autopista del Itata'], 'mapUrl': 'https://www.mapquestapi.com/staticmap/v5/map?key=NBRA305IVkuxUEwkz8zoXjBN6SeNyqPr&size=225,160&locations=-36.657710000000144,-72.45204000000001|marker-10||-36.656890000000146,-72.44968|marker-11||&center=-36.65730000000015,-72.45086&defaultMarker=none&zoom=15&session=AI8A5wcAAH4AAAAAAAAAEgAAAAIBAAB42mNgYmRgZGJgYGDPSC1KtUrOXTu9XhTIZeA2VW\_kCsnn9bmw-LxqKpDeCqQZsACYxtSOXLBGKeGXjUzd0qG-8\_P3aIHoeUCaAQdYn-G46S0zkJHXL6C4xomBvZVHgIODgWqgAbuwIAMbgyMLhN2cxAAAsSggRVidKCk:car'}, {'index': 10, 'distance': 12.1441, 'narrative': 'Continue on 152 (Acceso Norte A Concepción). Go for 12.1 mi.', 'time': 697, 'direction': 8, 'directionName': 'East', 'signs': [], 'maneuverNotes': [], 'formattedTime': '00:11:37', 'transportMode': 'car', 'startPoint': {'lat': -36.656890000000146, 'lng': -72.44968}, 'turnType': 0, 'attributes': 0, 'iconUrl': '', 'streets': ['152', 'Acceso Norte A Concepción', 'Autopista del Itata'], 'mapUrl': 'https://www.mapquestapi.com/staticmap/v5/map?key=NBRA305IVkuxUEwkz8zoXjBN6SeNyqPr&size=225,160&locations=-36.656890000000146,-72.44968|marker-11||-36.65413000000014,-72.23370999999987|marker-12||&center=-36.65551000000014,-72.34169499999993&defaultMarker=none&zoom=9&session=AI8A5wcAAH4AAAAAAAAAEgAAAAIBAAB42mNgYmRgZGJgYGDPSC1KtUrOXTu9XhTIZeA2VW\_kCsnn9bmw-LxqKpDeCqQZsACYxtSOXLBGKeGXjUzd0qG-8\_P3aIHoeUCaAQdYn-G46S0zkJHXL6C4xomBvZVHgIODgWqgAbuwIAMbgyMLhN2cxAAAsSggRVidKCk:car'}, {'index': 11, 'distance': 233.3392, 'narrative': 'Take the left exit toward Al Norte onto 5 (Ruta del Maule). Go for 233 mi.', 'time': 14134, 'direction': 3, 'directionName': 'Northeast', 'signs': [], 'maneuverNotes': [], 'formattedTime': '03:55:34', 'transportMode': 'car', 'startPoint': {'lat': -36.65413000000014, 'lng': -72.23370999999987}, 'turnType': 15, 'attributes': 0, 'iconUrl': '', 'streets': ['5', 'Ruta del Maule', 'Vía Panamericana', 'Longitudinal Sur', 'Puente Nebuco', 'Puente Ñuble', 'Puente Menelhue', 'Puente Ninquihue', 'Puente Navotavo', 'Puente Gaona', 'Puente Quiliahua', 'Puente Bulí', 'Carretera Panamericana', 'Puente Perquilaquén', 'Puente Longaví', 'Puente Achibueno', 'Puente Ancoa 2', 'Puente Ancoa 1', 'Puente Putagan', 'Puente Quilipin', 'Puente Los Pinos', 'Puente Los Patos', 'Puente Cunaco', 'Puente Toledo', 'Puente Motor', 'Puente Vertientes', 'Puente Maule', 'Puente Piduco', 'Puente Pangue', 'Puente Chagres', 'Puente Río Claro', 'Ruta del Maipo', 'Puente Pirihuin', 'Puente Lontué', 'Puente Río Teno', 'Puente Peor es Nada', 'Puente Descarga No 2', 'Puente Descarga', 'Puente Tinguiririca', 'Puente Antivero', 'Puente Charquicán', 'Puente Rigolemu', 'Puente Tipaume', 'Autopista Bypass Rancagua', 'Puente Cachapoal', 'Puente La Cadena', 'Puente Tronco', 'Puente Peuco'], 'mapUrl': 'https://www.mapquestapi.com/staticmap/v5/map?key=NBRA305IVkuxUEwkz8zoXjBN6SeNyqPr&size=225,160&locations=-36.65413000000014,-72.23370999999987|marker-12||-33.77382999999982,-70.74360999999931|marker-13||&center=-35.21397999999998,-71.48865999999958&defaultMarker=none&zoom=5&session=AI8A5wcAAH4AAAAAAAAAEgAAAAIBAAB42mNgYmRgZGJgYGDPSC1KtUrOXTu9XhTIZeA2VW\_kCsnn9bmw-LxqKpDeCqQZsACYxtSOXLBGKeGXjUzd0qG-8\_P3aIHoeUCaAQdYn-G46S0zkJHXL6C4xomBvZVHgIODgWqgAbuwIAMbgyMLhN2cxAAAsSggRVidKCk:car'}, {'index': 12, 'distance': 5.8856, 'narrative': 'Continue on 5. Go for 5.9 mi.', 'time': 395, 'direction': 1, 'directionName': 'North', 'signs': [], 'maneuverNotes': [], 'formattedTime': '00:06:35', 'transportMode': 'car', 'startPoint': {'lat': -33.77382999999982, 'lng': -70.74360999999931}, 'turnType': 0, 'attributes': 0, 'iconUrl': '', 'streets': ['5', 'Ruta del Maipo', 'Vía Panamericana', 'Longitudinal Sur', 'Puente Río Maipo'], 'mapUrl': 'https://www.mapquestapi.com/staticmap/v5/map?key=NBRA305IVkuxUEwkz8zoXjBN6SeNyqPr&size=225,160&locations=-33.77382999999982,-70.74360999999931|marker-13||-33.69136999999976,-70.72477999999933|marker-14||&center=-33.73259999999979,-70.73419499999932&defaultMarker=none&zoom=10&session=AI8A5wcAAH4AAAAAAAAAEgAAAAIBAAB42mNgYmRgZGJgYGDPSC1KtUrOXTu9XhTIZeA2VW\_kCsnn9bmw-LxqKpDeCqQZsACYxtSOXLBGKeGXjUzd0qG-8\_P3aIHoeUCaAQdYn-G46S0zkJHXL6C4xomBvZVHgIODgWqgAbuwIAMbgyMLhN2cxAAAsSggRVidKCk:car'}, {'index': 13, 'distance': 9.5051, 'narrative': 'Continue on 5 (Ruta del Maipo). Go for 9.5 mi.', 'time': 620, 'direction': 1, 'directionName': 'North', 'signs': [], 'maneuverNotes': [], 'formattedTime': '00:10:20', 'transportMode': 'car', 'startPoint': {'lat': -33.69136999999976, 'lng': -70.72477999999933}, 'turnType': 0, 'attributes': 0, 'iconUrl': '', 'streets': ['5', 'Ruta del Maipo', 'Vía Panamericana', 'Longitudinal Sur', 'Puente Río Maipo', 'Autopista Central'], 'mapUrl': 'https://www.mapquestapi.com/staticmap/v5/map?key=NBRA305IVkuxUEwkz8zoXjBN6SeNyqPr&size=225,160&locations=-33.69136999999976,-70.72477999999933|marker-14||-33.556579999999656,-70.7097699999993|marker-15||&center=-33.6239749999997,-70.71727499999932&defaultMarker=none&zoom=9&session=AI8A5wcAAH4AAAAAAAAAEgAAAAIBAAB42mNgYmRgZGJgYGDPSC1KtUrOXTu9XhTIZeA2VW\_kCsnn9bmw-LxqKpDeCqQZsACYxtSOXLBGKeGXjUzd0qG-8\_P3aIHoeUCaAQdYn-G46S0zkJHXL6C4xomBvZVHgIODgWqgAbuwIAMbgyMLhN2cxAAAsSggRVidKCk:car'}, {'index': 14, 'distance': 12.9456, 'narrative': 'Keep left onto 74 (Autopista Central) toward General Velásquez. Go for 12.9 mi.', 'time': 963, 'direction': 1, 'directionName': 'North', 'signs': [], 'maneuverNotes': [], 'formattedTime': '00:16:03', 'transportMode': 'car', 'startPoint': {'lat': -33.556579999999656, 'lng': -70.7097699999993}, 'turnType': 17, 'attributes': 0, 'iconUrl': '', 'streets': ['74', 'Autopista Central', 'Vía Panamericana'], 'mapUrl': 'https://www.mapquestapi.com/staticmap/v5/map?key=NBRA305IVkuxUEwkz8zoXjBN6SeNyqPr&size=225,160&locations=-33.556579999999656,-70.7097699999993|marker-15||-33.37697999999952,-70.695319999999|marker-16||&center=-33.46677999999959,-70.70254499999915&defaultMarker=none&zoom=9&session=AI8A5wcAAH4AAAAAAAAAEgAAAAIBAAB42mNgYmRgZGJgYGDPSC1KtUrOXTu9XhTIZeA2VW\_kCsnn9bmw-LxqKpDeCqQZsACYxtSOXLBGKeGXjUzd0qG-8\_P3aIHoeUCaAQdYn-G46S0zkJHXL6C4xomBvZVHgIODgWqgAbuwIAMbgyMLhN2cxAAAsSggRVidKCk:car'}, {'index': 15, 'distance': 286.2906, 'narrative': 'Continue on 5 (Autopista Central). Go for 286 mi.', 'time': 15929, 'direction': 1, 'directionName': 'North', 'signs': [], 'maneuverNotes': [], 'formattedTime': '04:25:29', 'transportMode': 'car', 'startPoint': {'lat': -33.37697999999952, 'lng': -70.695319999999}, 'turnType': 0, 'attributes': 0, 'iconUrl': '', 'streets': ['5', 'Autopista Central', 'Vía Panamericana', 'Autopista Aconcagua', 'Longitudinal Norte', 'Puente Chacabuco', 'Puente Rungue', 'Puente Montenegro', 'Puente Las Chilcas', 'Túnel La Calavera', 'Puente Paso Superior La Calavera', 'Puente Paso Superior Ocoa', 'Puente Aconcagua', 'Paso Superior Romeral', 'Puente Paso Superior Cuatro Esquinas', 'Puente El Litre', 'Puente Nogales', 'Puente El Cobre', 'Túnel El Melón II', 'Puente Quebradilla', 'Puente Talaguén', 'Puente Jaururo', 'Puente Pullally', 'Puente Petorca', 'Puente Longotoma', 'Puente Huaquén', 'Puente La Ballena', 'Puente El Chivato', 'Puente Huentelauquén', 'Puente El Teniente', 'Puente Limarí', 'Puente Camarones N1', 'Puente Camarones N2', 'Puente Lagunillas', 'La Hacienda', 'Totoralillo', 'Punte Culebrón', 'Avenida Juan Bohon'], 'mapUrl': 'https://www.mapquestapi.com/staticmap/v5/map?key=NBRA305IVkuxUEwkz8zoXjBN6SeNyqPr&size=225,160&locations=-33.37697999999952,-70.695319999999|marker-16||-29.90216999999956,-71.25642999999913|marker-17||&center=-31.63957499999954,-70.97587499999906&defaultMarker=none&zoom=5&session=AI8A5wcAAH4AAAAAAAAAEgAAAAIBAAB42mNgYmRgZGJgYGDPSC1KtUrOXTu9XhTIZeA2VW\_kCsnn9bmw-LxqKpDeCqQZsACYxtSOXLBGKeGXjUzd0qG-8\_P3aIHoeUCaAQdYn-G46S0zkJHXL6C4xomBvZVHgIODgWqgAbuwIAMbgyMLhN2cxAAAsSggRVidKCk:car'}, {'index': 16, 'distance': 0.2063, 'narrative': 'Turn right onto Avenida Brasil. Go for 0.2 mi.', 'time': 42, 'direction': 3, 'directionName': 'Northeast', 'signs': [], 'maneuverNotes': [], 'formattedTime': '00:00:42', 'transportMode': 'car', 'startPoint': {'lat': -29.90216999999956, 'lng': -71.25642999999913}, 'turnType': 2, 'attributes': 0, 'iconUrl': '', 'streets': ['Avenida Brasil'], 'mapUrl': 'https://www.mapquestapi.com/staticmap/v5/map?key=NBRA305IVkuxUEwkz8zoXjBN6SeNyqPr&size=225,160&locations=-29.90216999999956,-71.25642999999913|marker-17||-29.90093999999956,-71.25386999999914|marker-18||&center=-29.90155499999956,-71.25514999999913&defaultMarker=none&zoom=15&session=AI8A5wcAAH4AAAAAAAAAEgAAAAIBAAB42mNgYmRgZGJgYGDPSC1KtUrOXTu9XhTIZeA2VW\_kCsnn9bmw-LxqKpDeCqQZsACYxtSOXLBGKeGXjUzd0qG-8\_P3aIHoeUCaAQdYn-G46S0zkJHXL6C4xomBvZVHgIODgWqgAbuwIAMbgyMLhN2cxAAAsSggRVidKCk:car'}, {'index': 17, 'distance': 0.1684, 'narrative': 'Continue on Calle Brasil. Go for 0.2 mi.', 'time': 41, 'direction': 8, 'directionName': 'East', 'signs': [], 'maneuverNotes': [], 'formattedTime': '00:00:41', 'transportMode': 'car', 'startPoint': {'lat': -29.90093999999956, 'lng': -71.25386999999914}, 'turnType': 0, 'attributes': 0, 'iconUrl': '', 'streets': ['Calle Brasil'], 'mapUrl': 'https://www.mapquestapi.com/staticmap/v5/map?key=NBRA305IVkuxUEwkz8zoXjBN6SeNyqPr&size=225,160&locations=-29.90093999999956,-71.25386999999914|marker-18||-29.900879999999564,-71.25103999999914|marker-19||&center=-29.900909999999563,-71.25245499999915&defaultMarker=none&zoom=15&session=AI8A5wcAAH4AAAAAAAAAEgAAAAIBAAB42mNgYmRgZGJgYGDPSC1KtUrOXTu9XhTIZeA2VW\_kCsnn9bmw-LxqKpDeCqQZsACYxtSOXLBGKeGXjUzd0qG-8\_P3aIHoeUCaAQdYn-G46S0zkJHXL6C4xomBvZVHgIODgWqgAbuwIAMbgyMLhN2cxAAAsSggRVidKCk:car'}, {'index': 18, 'distance': 0.1056, 'narrative': 'Turn right onto Calle Los Carrera. Go for 0.1 mi.', 'time': 36, 'direction': 4, 'directionName': 'South', 'signs': [], 'maneuverNotes': [], 'formattedTime': '00:00:36', 'transportMode': 'car', 'startPoint': {'lat': -29.900879999999564, 'lng': -71.25103999999914}, 'turnType': 2, 'attributes': 0, 'iconUrl': '', 'streets': ['Calle Los Carrera'], 'mapUrl': 'https://www.mapquestapi.com/staticmap/v5/map?key=NBRA305IVkuxUEwkz8zoXjBN6SeNyqPr&size=225,160&locations=-29.900879999999564,-71.25103999999914|marker-19||-29.902389999999563,-71.25123999999914|marker-20||&center=-29.901634999999565,-71.25113999999914&defaultMarker=none&zoom=16&session=AI8A5wcAAH4AAAAAAAAAEgAAAAIBAAB42mNgYmRgZGJgYGDPSC1KtUrOXTu9XhTIZeA2VW\_kCsnn9bmw-LxqKpDeCqQZsACYxtSOXLBGKeGXjUzd0qG-8\_P3aIHoeUCaAQdYn-G46S0zkJHXL6C4xomBvZVHgIODgWqgAbuwIAMbgyMLhN2cxAAAsSggRVidKCk:car'}, {'index': 19, 'distance': 0, 'narrative': 'Arrive at Calle Los Carrera.', 'time': 0, 'direction': 0, 'directionName': 'None', 'signs': [], 'maneuverNotes': [], 'formattedTime': '00:00:00', 'transportMode': 'car', 'startPoint': {'lat': -29.902389999999563, 'lng': -71.25123999999914}, 'turnType': 0, 'attributes': 0, 'iconUrl': '', 'streets': []}]}], 'options': {'routeType': 'FASTEST', 'narrativeType': 'text', 'enhancedNarrative': False, 'walkingSpeed': -1, 'highwayEfficiency': 22, 'avoids': False, 'generalize': -1, 'shapeFormat': 'raw', 'unit': 'M', 'locale': 'en\_US', 'useTraffic': False, 'timeType': 0, 'dateType': 0, 'doReverseGeocode': True, 'manMaps': True, 'sideOfStreetDisplay': True}, 'boundingBox': {'ul': {'lat': -29.900879999999564, 'lng': -73.07847000000004}, 'lr': {'lat': -36.82996000000001, 'lng': -70.68528999999907}}, 'name': '5', 'maxRoutes': '', 'locations': [{'street': '', 'adminArea6': '', 'adminArea6Type': 'Neighborhood', 'adminArea5': 'Concepción', 'adminArea5Type': 'City', 'adminArea4': 'Concepción', 'adminArea4Type': 'County', 'adminArea3': 'BI', 'adminArea3Type': 'State', 'adminArea1': 'CL', 'adminArea1Type': 'Country', 'postalCode': '', 'geocodeQualityCode': 'A5XAX', 'geocodeQuality': 'CITY', 'dragPoint': False, 'sideOfStreet': 'N', 'linkId': '0', 'unknownInput': '', 'type': 's', 'latLng': {'lat': -36.82815, 'lng': -73.0512}, 'displayLatLng': {'lat': -36.82815, 'lng': -73.0512}, 'mapUrl': ''}, {'street': '', 'adminArea6': '', 'adminArea6Type': 'Neighborhood', 'adminArea5': 'La Serena', 'adminArea5Type': 'City', 'adminArea4': 'Elqui', 'adminArea4Type': 'County', 'adminArea3': 'CO', 'adminArea3Type': 'State', 'adminArea1': 'CL', 'adminArea1Type': 'Country', 'postalCode': '', 'geocodeQualityCode': 'A5XAX', 'geocodeQuality': 'CITY', 'dragPoint': False, 'sideOfStreet': 'N', 'linkId': '0', 'unknownInput': '', 'type': 's', 'latLng': {'lat': -29.90239, 'lng': -71.25124}, 'displayLatLng': {'lat': -29.90239, 'lng': -71.25124}, 'mapUrl': ''}], 'locationSequence': [0, 1]}, 'info': {'statuscode': 0, 'copyright': {'text': '© 2022 MapQuest, Inc.', 'imageUrl': 'http://api.mqcdn.com/res/mqlogo.gif', 'imageAltText': '© 2022 MapQuest, Inc.'}, 'messages': []}} |
| Consulta Quinta región |
| orig\_benjamincornejo = "villa alemana, chile"  dest\_benjamincornejo = "quilpue, chile"  devasc@labvm:~/labs/devnet-src/mapquest$ python3 /home/devasc/labs/devnet-src/mapquest/mapquest\_parse-json\_1.py   * + - 1. {'route': {'sessionId': 'AIIA5wcAAH4AAAAAAAAAEgAAAJUAAAB42mOYysDAyMTAwMCekVqUapWc-8Ok0xjIZWiU\_9HI9cDe3pe\_rkHjMZC-UtugwYAFwDS2RC4QBfE5C8PqmNJvGvpG2jRoZANpORvsGkHArTvjeBOQdpBmUGDQALljAhMTgwMDAwAP7x0xQFJ\_1Q:car', 'realTime': 1036, 'distance': 4.4596, 'time': 635, 'formattedTime': '00:10:35', 'hasHighway': False, 'hasTollRoad': False, 'hasBridge': True, 'hasSeasonalClosure': False, 'hasTunnel': False, 'hasFerry': False, 'hasUnpaved': False, 'hasTimedRestriction': False, 'hasCountryCross': False, 'legs': [{'index': 0, 'hasTollRoad': False, 'hasHighway': False, 'hasBridge': True, 'hasUnpaved': False, 'hasTunnel': False, 'hasSeasonalClosure': False, 'hasFerry': False, 'hasCountryCross': False, 'hasTimedRestriction': False, 'distance': 4.4596, 'time': 1036, 'formattedTime': '00:17:16', 'origIndex': 0, 'origNarrative': '', 'destIndex': 0, 'destNarrative': '', 'maneuvers': [{'index': 0, 'distance': 1.5783, 'narrative': 'Head west on Avenida Valparaíso. Go for 1.6 mi.', 'time': 324, 'direction': 7, 'directionName': 'West', 'signs': [], 'maneuverNotes': [], 'formattedTime': '00:05:24', 'transportMode': 'car', 'startPoint': {'lat': -33.044175, 'lng': -71.37131}, 'turnType': 0, 'attributes': 0, 'iconUrl': '', 'streets': ['Avenida Valparaíso'], 'mapUrl': 'https://www.mapquestapi.com/staticmap/v5/map?key=NBRA305IVkuxUEwkz8zoXjBN6SeNyqPr&size=225,160&locations=-33.044175,-71.37131|marker-1||-33.046160000000036,-71.39833000000002|marker-2||&center=-33.04516750000002,-71.38482&defaultMarker=none&zoom=12&session=AIIA5wcAAH4AAAAAAAAAEgAAAJUAAAB42mOYysDAyMTAwMCekVqUapWc-8Ok0xjIZWiU\_9HI9cDe3pe\_rkHjMZC-UtugwYAFwDS2RC4QBfE5C8PqmNJvGvpG2jRoZANpORvsGkHArTvjeBOQdpBmUGDQALljAhMTgwMDAwAP7x0xQFJ\_1Q:car'}, {'index': 1, 'distance': 2.5706, 'narrative': 'Continue on Avenida Ramón Freire. Go for 2.6 mi.', 'time': 550, 'direction': 7, 'directionName': 'West', 'signs': [], 'maneuverNotes': [], 'formattedTime': '00:09:10', 'transportMode': 'car', 'startPoint': {'lat': -33.046160000000036, 'lng': -71.39833000000002}, 'turnType': 0, 'attributes': 0, 'iconUrl': '', 'streets': ['Avenida Ramón Freire'], 'mapUrl': 'https://www.mapquestapi.com/staticmap/v5/map?key=NBRA305IVkuxUEwkz8zoXjBN6SeNyqPr&size=225,160&locations=-33.046160000000036,-71.39833000000002|marker-2||-33.04620000000002,-71.44071000000004|marker-3||&center=-33.04618000000003,-71.41952000000003&defaultMarker=none&zoom=11&session=AIIA5wcAAH4AAAAAAAAAEgAAAJUAAAB42mOYysDAyMTAwMCekVqUapWc-8Ok0xjIZWiU\_9HI9cDe3pe\_rkHjMZC-UtugwYAFwDS2RC4QBfE5C8PqmNJvGvpG2jRoZANpORvsGkHArTvjeBOQdpBmUGDQALljAhMTgwMDAwAP7x0xQFJ\_1Q:car'}, {'index': 2, 'distance': 0.0304, 'narrative': 'Continue on Avenida Blanco Encalada. Go for 161 ft.', 'time': 12, 'direction': 7, 'directionName': 'West', 'signs': [], 'maneuverNotes': [], 'formattedTime': '00:00:12', 'transportMode': 'car', 'startPoint': {'lat': -33.04620000000002, 'lng': -71.44071000000004}, 'turnType': 0, 'attributes': 0, 'iconUrl': '', 'streets': ['Avenida Blanco Encalada'], 'mapUrl': 'https://www.mapquestapi.com/staticmap/v5/map?key=NBRA305IVkuxUEwkz8zoXjBN6SeNyqPr&size=225,160&locations=-33.04620000000002,-71.44071000000004|marker-3||-33.046240000000026,-71.44123000000003|marker-4||&center=-33.04622000000002,-71.44097000000004&defaultMarker=none&zoom=16&session=AIIA5wcAAH4AAAAAAAAAEgAAAJUAAAB42mOYysDAyMTAwMCekVqUapWc-8Ok0xjIZWiU\_9HI9cDe3pe\_rkHjMZC-UtugwYAFwDS2RC4QBfE5C8PqmNJvGvpG2jRoZANpORvsGkHArTvjeBOQdpBmUGDQALljAhMTgwMDAwAP7x0xQFJ\_1Q:car'}, {'index': 3, 'distance': 0.0652, 'narrative': 'Turn left onto Calle Diego Portales. Go for 344 ft.', 'time': 19, 'direction': 6, 'directionName': 'Southwest', 'signs': [], 'maneuverNotes': [], 'formattedTime': '00:00:19', 'transportMode': 'car', 'startPoint': {'lat': -33.046240000000026, 'lng': -71.44123000000003}, 'turnType': 6, 'attributes': 0, 'iconUrl': '', 'streets': ['Calle Diego Portales'], 'mapUrl': 'https://www.mapquestapi.com/staticmap/v5/map?key=NBRA305IVkuxUEwkz8zoXjBN6SeNyqPr&size=225,160&locations=-33.046240000000026,-71.44123000000003|marker-4||-33.04697000000002,-71.44194000000003|marker-5||&center=-33.04660500000003,-71.44158500000003&defaultMarker=none&zoom=16&session=AIIA5wcAAH4AAAAAAAAAEgAAAJUAAAB42mOYysDAyMTAwMCekVqUapWc-8Ok0xjIZWiU\_9HI9cDe3pe\_rkHjMZC-UtugwYAFwDS2RC4QBfE5C8PqmNJvGvpG2jRoZANpORvsGkHArTvjeBOQdpBmUGDQALljAhMTgwMDAwAP7x0xQFJ\_1Q:car'}, {'index': 4, 'distance': 0.1125, 'narrative': 'Turn right onto Calle Claudio Vicuña. Go for 0.1 mi.', 'time': 72, 'direction': 2, 'directionName': 'Northwest', 'signs': [], 'maneuverNotes': [], 'formattedTime': '00:01:12', 'transportMode': 'car', 'startPoint': {'lat': -33.04697000000002, 'lng': -71.44194000000003}, 'turnType': 2, 'attributes': 0, 'iconUrl': '', 'streets': ['Calle Claudio Vicuña'], 'mapUrl': 'https://www.mapquestapi.com/staticmap/v5/map?key=NBRA305IVkuxUEwkz8zoXjBN6SeNyqPr&size=225,160&locations=-33.04697000000002,-71.44194000000003|marker-5||-33.04575000000003,-71.44322000000003|marker-6||&center=-33.04636000000002,-71.44258000000002&defaultMarker=none&zoom=16&session=AIIA5wcAAH4AAAAAAAAAEgAAAJUAAAB42mOYysDAyMTAwMCekVqUapWc-8Ok0xjIZWiU\_9HI9cDe3pe\_rkHjMZC-UtugwYAFwDS2RC4QBfE5C8PqmNJvGvpG2jRoZANpORvsGkHArTvjeBOQdpBmUGDQALljAhMTgwMDAwAP7x0xQFJ\_1Q:car'}, {'index': 5, 'distance': 0.1025, 'narrative': 'Continue on Calle Irarrázaval. Go for 0.1 mi.', 'time': 59, 'direction': 7, 'directionName': 'West', 'signs': [], 'maneuverNotes': [], 'formattedTime': '00:00:59', 'transportMode': 'car', 'startPoint': {'lat': -33.04575000000003, 'lng': -71.44322000000003}, 'turnType': 0, 'attributes': 0, 'iconUrl': '', 'streets': ['Calle Irarrázaval'], 'mapUrl': 'https://www.mapquestapi.com/staticmap/v5/map?key=NBRA305IVkuxUEwkz8zoXjBN6SeNyqPr&size=225,160&locations=-33.04575000000003,-71.44322000000003|marker-6||-33.045585000000024,-71.44492000000004|marker-7||&center=-33.04566750000002,-71.44407000000004&defaultMarker=none&zoom=16&session=AIIA5wcAAH4AAAAAAAAAEgAAAJUAAAB42mOYysDAyMTAwMCekVqUapWc-8Ok0xjIZWiU\_9HI9cDe3pe\_rkHjMZC-UtugwYAFwDS2RC4QBfE5C8PqmNJvGvpG2jRoZANpORvsGkHArTvjeBOQdpBmUGDQALljAhMTgwMDAwAP7x0xQFJ\_1Q:car'}, {'index': 6, 'distance': 0, 'narrative': 'Arrive at Calle Irarrázaval.', 'time': 0, 'direction': 0, 'directionName': 'None', 'signs': [], 'maneuverNotes': [], 'formattedTime': '00:00:00', 'transportMode': 'car', 'startPoint': {'lat': -33.045585000000024, 'lng': -71.44492000000004}, 'turnType': 0, 'attributes': 0, 'iconUrl': '', 'streets': []}]}], 'options': {'routeType': 'FASTEST', 'narrativeType': 'text', 'enhancedNarrative': False, 'walkingSpeed': -1, 'highwayEfficiency': 22, 'avoids': False, 'generalize': -1, 'shapeFormat': 'raw', 'unit': 'M', 'locale': 'en\_US', 'useTraffic': False, 'timeType': 0, 'dateType': 0, 'doReverseGeocode': True, 'manMaps': True, 'sideOfStreetDisplay': True}, 'boundingBox': {'ul': {'lat': -33.042180000000016, 'lng': -71.44492000000004}, 'lr': {'lat': -33.04697000000002, 'lng': -71.37131}}, 'name': 'Avenida Ramón Freire and Avenida Valparaíso', 'maxRoutes': '', 'locations': [{'street': '', 'adminArea6': '', 'adminArea6Type': 'Neighborhood', 'adminArea5': 'Villa Alemana', 'adminArea5Type': 'City', 'adminArea4': 'Marga Marga', 'adminArea4Type': 'County', 'adminArea3': 'VS', 'adminArea3Type': 'State', 'adminArea1': 'CL', 'adminArea1Type': 'Country', 'postalCode': '', 'geocodeQualityCode': 'A5XAX', 'geocodeQuality': 'CITY', 'dragPoint': False, 'sideOfStreet': 'N', 'linkId': '0', 'unknownInput': '', 'type': 's', 'latLng': {'lat': -33.04417, 'lng': -71.37131}, 'displayLatLng': {'lat': -33.04417, 'lng': -71.37131}, 'mapUrl': ''}, {'street': '', 'adminArea6': '', 'adminArea6Type': 'Neighborhood', 'adminArea5': 'Quilpué', 'adminArea5Type': 'City', 'adminArea4': 'Marga Marga', 'adminArea4Type': 'County', 'adminArea3': 'VS', 'adminArea3Type': 'State', 'adminArea1': 'CL', 'adminArea1Type': 'Country', 'postalCode': '', 'geocodeQualityCode': 'A5XAX', 'geocodeQuality': 'CITY', 'dragPoint': False, 'sideOfStreet': 'N', 'linkId': '0', 'unknownInput': '', 'type': 's', 'latLng': {'lat': -33.04558, 'lng': -71.44492}, 'displayLatLng': {'lat': -33.04558, 'lng': -71.44492}, 'mapUrl': ''}], 'locationSequence': [0, 1]}, 'info': {'statuscode': 0, 'copyright': {'text': '© 2022 MapQuest, Inc.', 'imageUrl': 'http://api.mqcdn.com/res/mqlogo.gif', 'imageAltText': '© 2022 MapQuest, Inc.'}, 'messages': []}} |

### Imprima la URL y compruebe el estado de la solicitud JSON.

Ahora que sabe que la solicitud JSON está funcionando, puede agregar algo más de funcionalidad a la aplicación.

* + - 1. Guarde su script como **mapquest\_parse-json\_2.py**.
      2. Elimine la instrucción **print (json\_data)** ya que no necesita probar que la solicitud tiene el formato correcto.
      3. Agregue las declaraciones a continuación, que hará lo siguiente:
* Imprima la URL construida para que el usuario pueda ver la solicitud exacta realizada por la aplicación.
* Analizar los datos JSON para obtener el valor de **código de estado**.
* Inicie un bucle **if** que comprueba si hay una llamada exitosa, que se indica con un valor devuelto de 0. Agregue una instrucción print para mostrar el valor de **código de estado** y su significado. El **\ n** agrega una línea en blanco debajo de la salida.

Más adelante en este laboratorio, agregará instrucciones **elif** y **else** para diferentes valores de **código de estado**.

print("URL: " + (url))

json\_data = requests.get(url).json()

json\_status = json\_data["info"]["statuscode"]

if json\_status == 0:

print("API Status: " + str(json\_status) + " = A successful route call.\n")

### Comandos de impresión de estado e URL de prueba.

* + - 1. El ejemplo aquí utiliza los siguientes parámetros.

orig = “Washington, D.C.”

dest = “Baltimore, Md”

* + - 1. Guarde y ejecute su script **mapquest\_parse-json\_2.py** y verifique que funcione. Solucione el problema de su código, si fuera necesario. Usted debería de obtener un resultado similar al siguiente. Observe que su clave está incrustada en la solicitud de URL.

devasc@labvm:~/labs/devnet-src/mapquest$ **python3 mapquest\_parse-json\_2.py**

URL: https://www.mapquestapi.com/directions/v2/route?key=fZadaFOY22VIEEemZcBFfxl5vjSXIPpZ&from=Washington%2C+D.C.&to=Baltimore%2C+Md

API Status: 0 = A successful route call.

devasc@labvm:~/labs/devnet-src/mapquest$

### Añadir entrada de usuario para la ubicación inicial y el destino.

Ha utilizado valores estáticos para las variables de ubicación. Sin embargo, la aplicación requiere que el usuario los introduzca. Complete los siguientes pasos para actualizar su aplicación:

* + - 1. Guarde su script como **mapquest\_parse-json\_3.py**.
      2. Elimine las variables **orig** y **dest** actuales.
      3. Vuelva a escribir el **orig** y **dest** para estar dentro de un bucle while en el que solicita la entrada del usuario para la ubicación inicial y el destino. El bucle while permite al usuario continuar realizando solicitudes para diferentes direcciones. Agregue el siguiente código, resaltado a continuación, después del parámetro clave. Asegúrese de que todo el código restante esté sangrado correctamente dentro del bucle while.

import urllib.parse

import requests

main\_api = "https://www.mapquestapi.com/directions/v2/route?"

key = "fZadaFOY22VIEEemZcBFfxl5vjSXIPpZ"

while True:

orig = input("Starting Location: ")

dest = input("Destination: ")

url = main\_api + urllib.parse.urlencode({"key": key, "from":orig, "to":dest})

print("URL: " + (url))

json\_data = requests.get(url).json()

json\_status = json\_data["info"]["statuscode"]

if json\_status == 0:

print("API Status: " + str(json\_status) + " = A successful route call.\n")

### Probar la funcionalidad de entrada del usuario.

Ejecute su script **mapquest\_parse-json\_3.py** y verifique que funcione. Solucione el problema de su código, si es necesario. Debería obtener una salida similar a la que se muestra a continuación. Para finalizar el programa, escriba **Ctrl+C**. Obtendrá un error de **KeyboardInterrupt** como se muestra en la salida a continuación. Para detener la aplicación con más gracia en ella, agregará la funcionalidad de **BREAK** en el siguiente paso.

devasc@labvm:~/labs/devnet-src/mapquest$ **python3 mapquest\_parse-json\_3.py**

Starting Location: Washington, D.C.

Destination: Baltimore, Md

URL: https://www.mapquestapi.com/directions/v2/route?key=fZadaFOY22VIEEemZcBFfxl5vjSXIPpZ&from=Washington%2C+D.C.&to=Baltimore%2C+Md

API Status: 0 = A successful route call.

Starting Location: **^C**

Traceback (most recent call last):

File "mapquest\_parse-json\_3.py", line 9, in <module>

orig = input("Starting Location: ")

KeyboardInterrupt

devasc@labvm:~/labs/devnet-src/mapquest$

### Agregue la funcionalidad para salir a la aplicación.

En lugar de forzar a la aplicación a salir con una interrupción del teclado, agregará la capacidad para que el usuario escriba **q** o **quit** como palabras clave para salir de la aplicación. Complete los siguientes pasos para actualizar su aplicación:

* + - 1. Guarde su script como **mapquest\_parse-json\_4.py**.
      2. Agregue una instrucción if después de cada variable de ubicación para comprobar si el usuario ingresa **q** o **quit**, como se muestra a continuación.

while True:

orig = input("Starting Location: ")

if orig == "quit" or orig == "q":

break

dest = input("Destination: ")

if dest == "quit" or dest == "q":

break

### Pruebe la funcionalidad de salir.

Ejecute el script **mapquest\_parse-json\_4.py** cuatro veces para probar cada variable de ubicación. Verifique que tanto **quit** y **q** terminen la aplicación cuando lo agregue a la Ubicación de origen y luego pruebe cuando consulte la Ubicación de destino. Solucione el problema de su código, si fuera necesario. Usted debería de obtener un resultado similar al siguiente.

Agregue evidencia de las 4 interrupciones del script.

|  |
| --- |
|  |
|  |
|  |
|  |

Al Script anterior de interrupción agregue además la instrucción **salir , ok , OK,** agregue evidencia del cambio e interrumpa nuevamente usando estas 3 opciones.

|  |
| --- |
|  |
|  |
|  |

### Mostrar los datos JSON en JSONView.

El navegador Chromium de la máquina virtual DEVASC incluye la extensión JsonView. Puede usar esto para ver un objeto JSON en un formato legible, coloreado y plegable.

* + - 1. Ejecute su **mapquest\_parse-json\_4.py** de nuevo y copie el código devuelto para la URL. No utilice la URL de este ejemplo ya que la **key** es personal.
      2. Su resultado incluirá su clave API.

devasc@labvm:~/labs/devnet-src/mapquest$ **python3 mapquest\_parse-json\_4.py**

Starting Location: **Washington, D.C.**

Destination: **Baltimore, Md**

URL: https://www.mapquestapi.com/directions/v2/route?key=fZadaFOY22VIEEemZcBFfxl5vjSXIPpZ&from=Washington%2C+D.C.&to=Baltimore%2C+Md

API Status: 0 = A successful route call.

Starting Location: **quit**

devasc@labvm:~/labs/devnet-src/mapquest$

* + - 1. Pegue la URL en el campo de direcciones del navegador Chromium
      2. Contraer los datos de JsonView seleccionando el guión “**-**” antes de la **ruta**, verá que hay dos diccionarios raíz: **ruta** e **información**.

{

- route:{

hasTollRoad: false,

hasBridge: true,

<output omitted>

Verá que hay dos diccionarios root: **rout** e **info**. Observe que la **información** tiene la clave/valor de **código de estado emparejado** utilizada en su código.

{

+ route: {},

- info: {

statuscode: 0,

- copyright: {

imageAltText: "© 2019 MapQuest, Inc.",

imageUrl: "http://api.mqcdn.com/res/mqlogo.gif",

text: "© 2019 MapQuest, Inc."

},

messages: [ ]

}

}

* + - 1. Expanda el diccionario de **ruta** (haga clic en el signo más “**+**” antes de la **route**) e investigue los datos enriquecidos. Hay valores que indican si la ruta tiene carreteras de peaje, puentes, túneles, autopistas, cierres o cruces hacia otros países. También deberías ver los valores de distancia, el tiempo total que tardará el viaje y el consumo de combustible. Para analizar y mostrar estos datos en su aplicación, debe especificar el diccionario de **route** y, a continuación, seleccionar el par key/value que desea imprimir. Hará un análisis del diccionario de ruta en la siguiente parte del laboratorio.
      2. Agregue evidencia de esta consulta del navegador Chromium

|  |
| --- |
| <https://www.mapquestapi.com/directions/v2/route?key=NBRA305IVkuxUEwkz8zoXjBN6SeNyqPr&from=vi%C3%B1a+del+mar%2C+chile&to=quilpue%2C+chile> |

## Actualice la aplicación Direcciones de MapQuest (MapQuest Directions) con más características

En esta parte, agregará características adicionales a la aplicación MapQuest Directions para proporcionar más información al usuario. Incluirá información de resumen del viaje y luego una lista de las direcciones analizadas desde el diccionario. Como paso final, agregará algunas comprobaciones básicas de errores para validar la entrada del usuario.

### Muestra información de resumen de viaje para incluir la duración, la distancia y el combustible utilizado. En este paso “route” “formattedTime” “distance” “maneuvers”.

* + - 1. Guarde su script como **mapquest\_parse-json\_5.py**.
      2. Debajo del comando API status **print**, agregue varias instrucciones de **print** que muestren las ubicaciones de origen y destino, así como las claves **FormattedTime**, **distance**.

Las instrucciones adicionales también incluyen instrucciones de impresión que mostrarán una línea doble antes de la siguiente solicitud de una ubicación inicial. Asegúrese de que estas instrucciones están incrustadas en la función while True.

if json\_status == 0:

print(“API Status: “ + str(json\_status) + “ = A successful route call.\n”)

print(“=============================================“)

print(“Directions from “ + (orig) + “ to “ + (dest))

print(“Trip Duration: “ + (json\_data[“route”][“formattedTime”]))

print (“Miles:” + str (json\_data [“route”] [“distance”]))

print(“=============================================“)

* + - 1. Guarde y ejecute **mapquest\_parse-json\_5.py** para ver la siguiente salida.

devasc@labvm:~/labs/devnet-src/mapquest$ **python3 mapquest\_parse-json\_5.py**

Lugar de inicio: **Washington, D.C.**

Destino: **Baltimore, Md**

URL: https://www.mapquestapi.com/directions/v2/route?key =FZADAFoy22VieeeEMZCBFXL5VJSXIPPZ&from=Washington%2C+D.C.&to=Baltimore%2C+MD

Estado de la API: 0 = Una llamada de ruta exitosa.

=============================================

Cómo llegar desde Washington, D.C. a Baltimore, Md

Duración del viaje: 00:49:29

Millas: 38.089

=============================================

Localización Inicial: **q**

devasc@labvm:~/labs/devnet-src/mapquest$

* + - 1. Por defecto, MapQuest utiliza el sistema imperial y no hay un parámetro de solicitud para cambiar los datos al sistema métrico. Por lo tanto, probablemente debería convertir su aplicación para mostrar los valores de métrica, transforme el resultado de millas a kilómetros. Evidencie el resultado en Kilometros.

|  |
| --- |
|  |

* + - 1. Agregue al script el consumo de combustible para los siguientes vehículos.
         1. City Car: Rendimiento Mixto 20 Km/L
         2. Sedan: Rendimiento Mixto 15 Km/L
         3. SUV: Rendimiento Mixto 12 Km/L
      2. Evidencie el resultado con los 3 formatos de vehículos y su consumo de cuanto combustible consumirán.

|  |
| --- |
|  |

* + - 1. Solicite al usuario ingresar un tipo de vehículo y debe mostrar el consumo para de acuerdo a la opción ingresada por el usuario. Agregue evidencia del resultado del script según la opción ingresada.

|  |
| --- |
|  |

* + - 1. Agregue el costo total de bencina, para esto puede utilizar cualquier categoría de bencina (93/95/97) y el valor de referencia puede ser un promedio de los 3. Agregue evidencia del resultado del script.

|  |
| --- |
|  |

* + - 1. Las posiciones decimales adicionales para Kilómetros y Combustible Usado no son útiles. Utilice el argumento **de formato “{: .2f}”**. Para formatear los valores flotantes a 2 lugares decimales antes de convertirlos en valores de cadena, como se muestra a continuación. Cada declaración debe estar en una línea. Agregue evidencia del resultado de consumo de combustible en litros con 2 decimales.

|  |
| --- |
|  |

### Inspeccione la lista de maniobras en los datos JSON.

* + - 1. Ahora está listo para mostrar las indicaciones paso a paso desde la ubicación inicial hasta el destino. Vuelva al navegador de Chromium donde vio anteriormente la salida en JsonView. Si usted cerró el navegador, copie la URL de la última vez que ejecutó el programa y péguela en la barra de direcciones del navegador.
      2. Dentro del diccionario de **ruta**, localice la lista de **tramos**. La lista de **reglas** incluye un diccionario grande con la mayoría de los datos JSON. Encuentre la lista de **maniobras** y colapse cada uno de los siete diccionarios dentro, como se muestra a continuación (haga clic en el **signo menos**”**-** “para cambiarlo a un **signo “+” más**). Si está utilizando diferentes ubicaciones, probablemente tendrá un número diferente de diccionarios de maniobra.

- legs: [

- {

hasTollRoad: false,

hasBridge: true,

destNarrative: "Proceed to BALTIMORE, MD.",

distance: 38.089,

hasTimedRestriction: false,

hasTunnel: false,

hasHighway: true,

index: 0,

formattedTime: "00:49:29",

origIndex: -1,

hasAccessRestriction: false,

hasSeasonalClosure: false,

hasCountryCross: false,

- roadGradeStrategy: [

[ ]

],

destIndex: 3,

time: 2969,

hasUnpaved: false,

origNarrative: "",

- maneuvers: [

+ {…},

+ {…},

+ {…},

+ {…},

+ {…},

+ {…},

+ {…}

],

hasFerry: false

}

],

- options: {

* + - 1. Expanda el primer diccionario de la lista de **maniobras**. Cada diccionario contiene una clave **narrativa** con un valor, como “Empezar hacia el norte...”, como se muestra a continuación. Debe analizar los datos JSON para extraer el valor para que la clave **narrativa** se muestre dentro de su aplicación.

- legs: [

- {

hasTollRoad: false,

hasBridge: true,

destNarrative: "Proceed to BALTIMORE, MD.",

distance: 38.089,

hasTimedRestriction: false,

hasTunnel: false,

hasHighway: true,

index: 0,

formattedTime: "00:49:29",

origIndex: -1,

hasAccessRestriction: false,

hasSeasonalClosure: false,

hasCountryCross: false,

- roadGradeStrategy: [

[ ]

],

destIndex: 3,

time: 2969,

hasUnpaved: false,

origNarrative: "",

- maneuvers: [

- {

distance: 0.792,

- streets: [

"6th St",

"US-50 E",

"US-1 N"

],

narrative: "Start out going north on 6th St/US-50E/US-1 N toward Pennsylvania Ave/US-1 Alt N.",

turnType: 0,

- startPoint: {

lng: -77.019913,

lat: 38.892063

},

index: 0,

formattedTime: "00:02:06",

directionName: "North",

maneuverNotes: [ ],

linkIds: [ ],

- signs: [

- {

extraText: "",

ext: "50",

type: 2,

<output omitted>

**Nota**: Se agregó el ajuste de Word para el valor de la narrativa a efectos de visualización.

### Agregue un bucle for para iterar a través de los datos JSON maniobras.

Complete los siguientes pasos para actualizar la aplicación y mostrar el valor de la clave **narrativa**. Lo hará creando un bucle for para iterar a través de la lista de maniobras, mostrando el valor narrativo de cada maniobra desde la ubicación inicial hasta el destino.

* + - 1. Guarde su script como **mapquest\_parse-json\_6.py**.
      2. Agregue un bucle “for”, resaltado a continuación, después de la segunda declaración de impresión de doble línea. El bucle for itera a través de cada lista de **maniobras** y hace lo siguiente:
         1. Imprima el valor **narrativo**.
         2. Valide los siguientes datos ya implementados en los pasos anteriores:

Convierta las millas en kilómetros con **\*1.61**.

Formatea el valor del kilómetro para imprimir sólo dos posiciones decimales con la función **“{:.2f}” .format**.

* + - 1. Agregue una instrucción de **impresión** que mostrará una línea doble antes de que la aplicación solicite otra ubicación inicial, como se muestra a continuación.

for each in json\_data["route"]["legs"][0]["maneuvers"]:

print((each["narrative"]) + " (" + str("{:.2f}".format((each["distance"])\*1.61) + " km)"))

print("=============================================\n")

### Actividad: pruebe la iteración de JSON.

Guarde y ejecute el script **mapquest\_parse-json\_6.py** para verificar que funciona. Solucione el problema de su código, si es necesario Usted debería de tener un resultado similar al siguiente

devasc@labvm:~/labs/devnet-src/mapquest$ **python3 mapquest\_parse-json\_6.py**

Localización inicial: **Washington, D.C.**

Destino: **Baltimore, Md**

Tipo de vehiculo: Según lo que usted haya creado.

URL: https://www.mapquestapi.com/directions/v2/route?key =FZADAFoy22VieeeEMZCBFXL5VJSXIPPZ&from=Washington%2C+D.C.&to=Baltimore%2C+MD

Estado de la API: 0 = Una llamada de ruta exitosa.

=============================================

Direcciones desde Washington, D.C. a Baltimore, Md

Duración del viaje: 00:49:29

Kilómetros: 61.32

Combustible utilizado (Litros): 6.24

Costo: $8.112

=============================================

Salga hacia el norte por la 6ª ST/US-50 E/US-1 N hacia Pennsylvania Ave/US-1 Alt N. (1,28 km)

Gire a la derecha en New York Ave/US-50 E. Y continúe por US-50 E (cruce hacia Maryland). (7.51 km)

Tome la salida Balt-Wash Parkway a la izquierda hacia Baltimore. (0.88 km)

Incorpórese en MD-295 N. (50.38 km)

Gire a la derecha hacia W Pratt St. (0,86 km)

Gire a la izquierda hacia S Calvert ST/MD-2. (0.43 km)

Bienvenido a BALTIMORE, MD. (0.00 km)

=============================================

Ubicación de inicio: **q**

devasc@labvm:~/labs/devnet-src/mapquest$

|  |
| --- |
| Agregue evidencia de la salida del script entre ciudades de Chile y entre ubicaciones de la Quinta región |