

Código para Iniciar Sesión

```
In [ ]: import requests
import json

# URL base de La API
base_url = "https://backend.electromovilidadenlinea.cl/"

# Endpoint para el inicio de sesión
sign_in_endpoint = "sign-in"

# URL completa para la solicitud de inicio de sesión
url = f"{base_url}{sign_in_endpoint}"

# Datos de inicio de sesión (ajusta según Los requerimientos de La API)
login_data = {
    "username": "11111111", #Rut de Usuario sin puntos ni guion (OPC o PSE asignado en front)
    "password": "password"
}

try:
    # Realizar la solicitud POST para iniciar sesión
    response = requests.post(url, json=login_data)

    # Verificar el código de estado de La respuesta
    if response.status_code == 200:
        print("Respuesta:", response.json())
        Token = response.json()['IdToken'] #Token de autenticación
    else:
        print("Respuesta:", response.text)

except requests.exceptions.RequestException as e:
    print(f"Error al realizar la solicitud: {e}")
```

Código para Actualizar Info. Cargadores (PATCH)

```
In [ ]: # URL completa para la solicitud PATCH evse
patch_url_evse = "https://backend.electromovilidadenlinea.cl/evses"

patch_data = { #agregar listado de campos modificables
    "evse_uid": 9999, #id Cargador
    "status": "OCUPADO", #"DISPONIBLE", "OCUPADO","NO DISPONIBLE", "FUERA DE LINEA"
    "payment_capabilities": ["APP", "DEBITO"], #"APP", "EFECTIVO", "DEBITO", "CREDITO", "PORTAL", "SUSCRIPCION"
    "activation_capabilities": ["APP", "QR"] #"APP", "RFID", "QR", "PORTAL"
    #"physical_reference": "Hola Mundo", #Referencia Fisica (interfaz)
    #"directions": "Cargador en el segundo piso del hotel" #Instrucciones para el cargador (interfaz)
}

# Headers para la solicitud, incluyendo el token de autenticación
headers = {
    "Authorization": f"Bearer {Token}",
    "Content-Type": "application/json"
}

try:
    # Realizar la solicitud PATCH
    response = requests.patch(patch_url_evse, json=patch_data, headers=headers)

    # Verificar el código de estado de La respuesta
    if response.status_code == 200:
        print("Actualización exitosa")
        print("Respuesta:", response.json())
    else:
        print(f"Error en la actualización. Código de estado: {response.status_code}")
        print("Respuesta:", response.text)

except requests.exceptions.RequestException as e:
    print(f"Error al realizar la solicitud: {e}")
```

Código para Actualizar Info. Conectores (PATCH)

```
In [ ]: # URL completa para la solicitud PATCH Connectors
patch_url_connectors = "https://backend.electromovilidadenlinea.cl/connectors"

patch_data = { #agregar listado de campos modificables
    "connector_id": 9999, #id Conector
    "status": "DISPONIBLE", #"DISPONIBLE", "OCUPADO","NO DISPONIBLE", "FUERA DE LINEA"
    #"soc":25, #SOC número de porcentaje de carga
    #"voltage": 10,
    #"amperage": 100,
    #"electric_power": 20
}

# Headers para la solicitud, incluyendo el token de autenticación
```

```
headers = {
    "Authorization": f"Bearer {Token}",
    "Content-Type": "application/json"
}

try:
    # Realizar la solicitud PATCH
    response = requests.patch(patch_url_connectors, json=patch_data, headers=headers)

    # Verificar el código de estado de la respuesta
    if response.status_code == 200:
        print("Respuesta:", response.json())
    else:
        print(f"Error en la actualización. Código de estado: {response.status_code}")
        print("Respuesta:", response.text)

except requests.exceptions.RequestException as e:
    print(f"Error al realizar la solicitud: {e}")
```

Código para Agregar Tarifa (POST)

```
In [ ]: # URL completa para la solicitud POST Tariffs
post_url_tariffs = "https://backend.electromovilidadenlinea.cl/tariffs"

post_data = { #agregar listado de campos modificables
    "order_number": 1,
    "tariff_alt_url": "https://www.sec.cl",
    "min_price": 0,
    "max_price": 15000,
    "connector_id": 9999, #Se debe identificar conector que posea asociada la tarifa
    "elements": {
        "price_components": [
            {
                "order_number": 1,
                "tariff_dimension": "CARGO FIJO", #TIEMPO, "CARGO FIJO", "ENERGIA", "POTENCIA"
                "price": 10.0,
            },
            {
                "order_number": 2,
                "tariff_dimension": "ENERGIA",
                "price": 250.0,
            }
        ]
    }
}

# Headers para la solicitud, incluyendo el token de autenticación
headers = {
    "Authorization": f"Bearer {Token}",
    "Content-Type": "application/json"
}

try:
    # Realizar la solicitud PATCH
    response = requests.post(post_url_tariffs, json=post_data, headers=headers)

    # Verificar el código de estado de la respuesta
    if response.status_code == 200:
        print("Respuesta:", response.json())
    else:
        print(f"Error en la actualización. Código de estado: {response.status_code}")
        print("Respuesta:", response.text)

except requests.exceptions.RequestException as e:
    print(f"Error al realizar la solicitud: {e}")
```

Código para Actualizar info. Tarifas (PATCH)

```
In [ ]: # URL completa para la solicitud PATCH Tariffs
patch_url_tariffs = "https://backend.electromovilidadenlinea.cl/tariffs"

patch_data = {
    "tariff_id": "999", #Se debe identificar tarifa (interfaz)
    "tariff_alt_url": "https://www.sec.cl",
    "min_price": 10,
    "max_price": 5000,
    "elements": {
        "price_components": [
            {
                "order_number": 1,
                "tariff_dimension": "TIEMPO",
                "price": 20,
            },
            {
                "order_number": 2,
                "tariff_dimension": "ENERGIA",
                "price": 150,
            }
        ]
    }
}
```

```

    ]
    }
}

# Headers para la solicitud, incluyendo el token de autenticación
headers = {
    "Authorization": f"Bearer {Token}",
    "Content-Type": "application/json"
}

try:
    # Realizar la solicitud PATCH
    response = requests.patch(patch_url_tariffs, json=patch_data, headers=headers)

    # Verificar el código de estado de la respuesta
    if response.status_code == 200:
        print("Respuesta:", response.json())
    else:
        print(f"Error en la actualización. Código de estado: {response.status_code}")
        print("Respuesta:", response.text)

except requests.exceptions.RequestException as e:
    print(f"Error al realizar la solicitud: {e}")

```

Código para Eliminar Tarifa (DEL)

```

In [ ]: # URL completa para la solicitud DEL Tariffs
DEL_url_tariffs = "https://backend.electromovilidadonlinea.cl/tariffs"

DEL_data = { #agregar listado de campos modificables
    "tariff_id": 999
}

# Headers para la solicitud, incluyendo el token de autenticación
headers = {
    "Authorization": f"Bearer {Token}",
    "Content-Type": "application/json"
}

try:
    # Realizar la solicitud DEL
    response = requests.delete(DEL_url_tariffs, json=DEL_data, headers=headers)

    # Verificar el código de estado de la respuesta
    if response.status_code == 200:
        print("Respuesta:", response.json())
    else:
        print(f"Error en la actualización. Código de estado: {response.status_code}")
        print("Respuesta:", response.text)

except requests.exceptions.RequestException as e:
    print(f"Error al realizar la solicitud: {e}")

```