



---

*LINKAR REST API con  
Javascript, .NET, JAVA, PHP,  
cURL.*

---

*Marzo 2020*

*by Kosday Solutions*



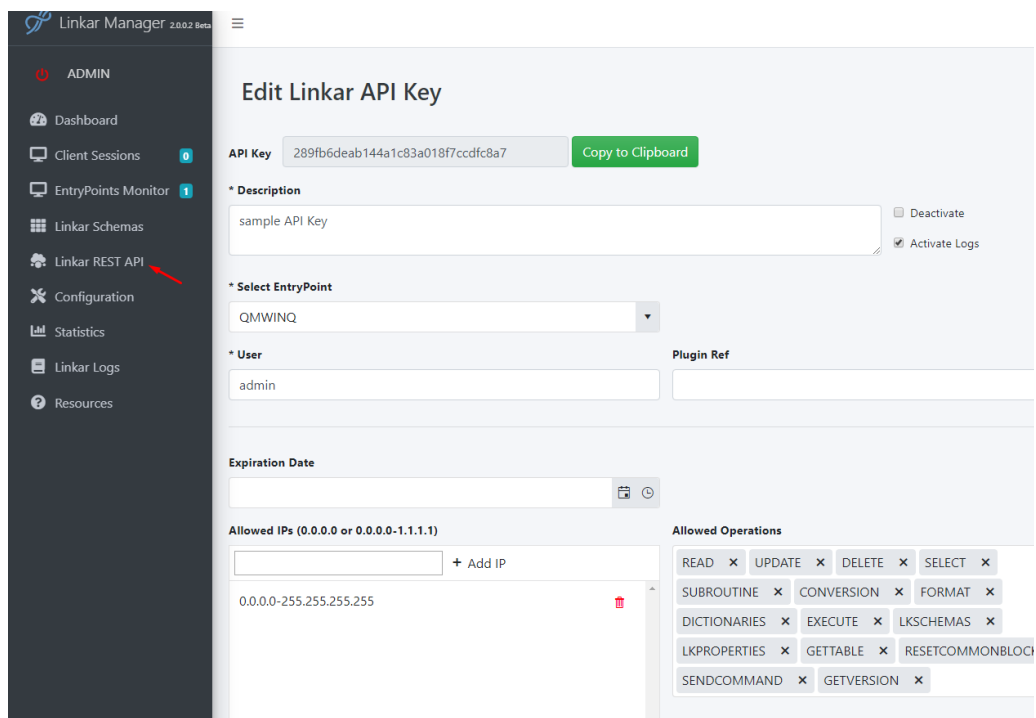
# LINKAR REST API con JavaScript, .NET, JAVA, PHP, cURL

Linkar REST API proporciona una interfaz para que las aplicaciones interactúen con su Linkar Suite enviando y recibiendo datos como objetos JSON o XML.

En este documento se mostrarán algunos ejemplos de llamadas a Linkar REST API desde los lenguajes de programación más populares.

Para realizar una llamada a la REST API necesitamos dos cosas:

- URL: esta url hace referencia a la dirección web (host y puerto) donde Linkar Manager está escuchando solicitudes http
- APIKey: esta APIKey se debe crear en Linkar Manager y definir en él las direcciones IPs y las operaciones permitidas para este cliente.



La solicitud se realizará a la URL de la operación que queremos realizar, por ejemplo, si deseamos ejecutar una READ, deberemos llamar a la URL de Linkar Manager y añadirle la operación a realizar (<http://yourdomain.com:11201/api/read>).

Linkar REST API utiliza todas las Plantillas de la función Sendcommand de Linkar:

Read, Update, New, Delete, Select, Subroutine, Conversion, Format, Dictionaries, Execute, LkSchemas, LkSchemasSqlMode, LkSchemasDictionaries, LkProperties,

# LINKAR REST API con JavaScript, .NET, JAVA, PHP, cURL

LkPropertiesSqlMode, LkPropertiesDictionaries, GetTable, GetTableSqlMode, GetTableDictionaries, GetTableNone, GetVersion, ResetCommonBlocks.

El servicio LINKAR MANAGER debe estar arrancado para poder utilizar Linkar REST API.

La llamada puede realizarse en dos formatos, JSON y XML, según el formato elegido para realizar la comunicación la llamada variará para que el servicio web la acepte.

Los cambios que deben realizarse según el formato escogido son los siguientes:

- ContentType:

JSON: su valor será "application/json".

XML: su valor será "text/xml".

- DataType: (no es necesario en todos los lenguajes)

JSON: su valor será "json".

XML: su valor será: "text".

- Data:

JSON: debe ser un objeto JSON convertido a texto plano.

XML: una cadena de texto con formato XML.

Estos son los requisitos para que la llamada sea aceptada, pero para que sea comprendida por el servicio los datos que se envían deben seguir una estructura predeterminada.

Le llamamos a esta estructura un LkCommand, si ejecutas el código de este ejemplo podrás ver con facilidad todos y cada uno de los comandos que se pueden ejecutar, pero vamos a tratar de explicar cómo se forman.

En XML debe de existir un nodo principal llamado <LkCommand>, dentro de este nodo se encontrará todo el contenido del comando.

En JSON el objeto principal es en sí mismo el LkCommand y no debe definirse como una propiedad dentro de un objeto superior

Esta es la forma correcta `{ "APIKey": "my_apikey", "Data": "..."}`

Esta sería una forma incorrecta `{ "LkCommand": { "APIKey": "my_apikey", "Data": "..."} }`

Como se puede ver en este último ejemplo un comando necesita contener dos variables para funcionar:

APIKey: la APIKey proporcionado por Linkar Manager.

Data: los datos de la operación que vamos a realizar.

## LINKAR REST API con JavaScript, .NET, JAVA, PHP, cURL

Data es en sí mismo otro paquete embutido dentro del LkCommand y es en este dónde le daremos a la operación los argumentos necesarios y opcionales para obtener el resultado deseado.

Cada operación tiene una plantilla concreta, en estos ejemplos se usa la plantilla de la operación READ.

Para consultar más información sobre todas las plantillas disponibles le recomendamos que consulte nuestro manual de ayuda.

# LINKAR REST API con JavaScript, .NET, JAVA, PHP, cURL

## Ejemplo de Read con JAVASCRIPT

```
var data = JSON.stringify({ APIKey : my_apikey, Data: ... });
$.ajax({
  url: "http://yourdomain.com:11201/api/read",
  contentType: "application/json",
  type: "POST",
  dataType: "json",
  data: data,
  success: function (result) {
  },
  error: function (error) {
  }
});
```

---

```
var data = '<LkCommand><APIKey>my_apikey</APIKey><Data>...</Data></LkCommand>';
$.ajax({
  url: "http://yourdomain.com:11201/api/read",
  contentType: "text/xml",
  type: "POST",
  dataType: "text",
  data: data,
  success: function (result) {
  },
  error: function (error) {
  }
});
```

# LINKAR REST API con JavaScript, .NET, JAVA, PHP, cURL

## Ejemplo de Read con .NET

```
HttpClient client = new HttpClient();
    HttpContent httpContent = new StringContent("{ \"APIKey\": \"my_apikey\",
    \"Data\": { \"CALCULATED\": \"False\", \"DICTIONARIES\": \"False\", \"CONVERSION\":
    \"False\", \"FORMAT_SPEC\": \"False\", \"ORIGINAL_RECORDS\": \"False\"
    , \"CUSTOM_VARS\": \"\", \"OUTPUT_FORMAT\": \"MV\", \"FILE_NAME\":
    \"LK.CUSTOMERS\", \"RECORD_IDS\": [ { \"ID\": \"2\" }, { \"ID\": \"3\" } ]
    , \"RECORD_DICTCLAUSE\": \"\" } }", Encoding.UTF8, "application/json");
    HttpResponseMessage response =
    client.PostAsync("http://yourdomain.com:11201/api/read", httpContent).Result;
    response.EnsureSuccessStatusCode();
    string responseBody = response.Content.ReadAsStringAsync().Result;
```

---

```
HttpClient client = new HttpClient();
    HttpContent httpContent = new
    StringContent("<LkCommand><APIKey>my_apikey</APIKey><Data><CALCULATED>False</C
    ALCULATED><DICTIONARIES>False</DICTIONARIES><CONVERSION>False</CONVERSION><F
    ORMAT_SPEC>False</FORMAT_SPEC><ORIGINAL_RECORDS>False</ORIGINAL_RECORDS><C
    USTOM_VARS></CUSTOM_VARS><OUTPUT_FORMAT>MV</OUTPUT_FORMAT><FILE_NAME
    >LK.CUSTOMERS</FILE_NAME><RECORD_IDS><ID>2</ID><ID>3</ID></RECORD_IDS><RECO
    RD_DICTCLAUSE></RECORD_DICTCLAUSE></Data></LkCommand>", Encoding.UTF8,
    "text/xml");
    HttpResponseMessage response =
    client.PostAsync("http://yourdomain.com:11201/api/read", httpContent).Result;
    response.EnsureSuccessStatusCode();
    string responseBody = response.Content.ReadAsStringAsync().Result;
```

# LINKAR REST API con JavaScript, .NET, JAVA, PHP, cURL

## Ejemplo de Read con JAVA

```
try
{
    URL url = new URL("http://yourdomain.com:11201/api/read");
    HttpURLConnection conn = (HttpURLConnection)
url.openConnection();
    conn.setDoOutput(true);
    conn.setRequestMethod("POST");
    conn.setRequestProperty("Content-Type", "application/json");
    String input = "{ \"APIKey\" : \"my_apikey\", \"Data\" : {
\"CALCULATED\" : \"False\", \"DICTIONARIES\" : \"False\", \"CONVERSION\" : \"False\",
\"FORMAT_SPEC\" : \"False\", \"ORIGINAL_RECORDS\" : \"False\", \"CUSTOM_VARS\" : \"\"
, \"OUTPUT_FORMAT\" : \"MV\", \"FILE_NAME\" : \"LK.CUSTOMERS\", \"RECORD_IDS\" : [ {
\"ID\" : \"2\", { \"ID\" : \"3\" } ], \"RECORD_DICTCLAUSE\" : \"\" } }";

    OutputStream os = conn.getOutputStream();
    os.write(input.getBytes());
    os.flush();

    if (conn.getResponseCode() ==
HttpURLConnection.HTTP_BAD_REQUEST)
    {
        throw new RuntimeException("Failed: HTTP error code: " +
conn.getResponseCode());
    }

    BufferedReader br = new BufferedReader(new
InputStreamReader(conn.getInputStream(), "UTF-8"));
    String output;
    System.out.println("Output from Server ... \n");
    while ((output = br.readLine()) != null ) {
        System.out.println(output);
    }
}
catch(MalformedURLException e) {
    e.printStackTrace();
}
catch(IOException e) {
    e.printStackTrace();
}
}
```

# LINKAR REST API con JavaScript, .NET, JAVA, PHP, cURL

```
try
    {
        URL url = new URL("http://yourdomain.com:11201/api/read");
        HttpURLConnection conn = (HttpURLConnection)
url.openConnection();
        conn.setDoOutput(true);
        conn.setRequestMethod("POST");
        conn.setRequestProperty("Content-Type", "text/xml");
        String input =
"<LkCommand><APIKey>my_apikey</APIKey><Data><CALCULATED>False</CALCULATED><D
ICTIONARIES>False</DICTIONARIES><CONVERSION>False</CONVERSION><FORMAT_SPEC>F
alse</FORMAT_SPEC><ORIGINAL_RECORDS>False</ORIGINAL_RECORDS><CUSTOM_VARS><
/CUSTOM_VARS><OUTPUT_FORMAT>MV</OUTPUT_FORMAT><FILE_NAME>LK.CUSTOMER
S</FILE_NAME><RECORD_IDS><ID>2</ID><ID>3</ID></RECORD_IDS><RECORD_DICTCLAU
SE></RECORD_DICTCLAUSE></Data></LkCommand>";

        OutputStream os = conn.getOutputStream();
        os.write(input.getBytes());
        os.flush();

        if (conn.getResponseCode() ==
HttpURLConnection.HTTP_BAD_REQUEST)
            {
                throw new RuntimeException("Failed: HTTP error code: " +
conn.getResponseCode());
            }

        BufferedReader br = new BufferedReader(new
InputStreamReader(conn.getInputStream(), "UTF-8"));
        String output;
        System.out.println("Output from Server ... \n");
        while ((output = br.readLine()) != null ) {
            System.out.println(output);
        }
    }
catch(MalformedURLException e) {
    e.printStackTrace();
}
catch(IOException e) {
    e.printStackTrace();
}
```



# LINKAR REST API con JavaScript, .NET, JAVA, PHP, cURL

## Ejemplo de Read con PHP

```
try
{
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS, "{ \"APIKey\": \"my_apikey\", \"Data\": {
    \"CALCULATED\": \"False\", \"DICTIONARIES\": \"False\", \"CONVERSION\": \"False\",
    \"FORMAT_SPEC\": \"False\", \"ORIGINAL_RECORDS\": \"False\", \"CUSTOM_VARS\": \"\"
    , \"OUTPUT_FORMAT\": \"MV\", \"FILE_NAME\": \"LK.CUSTOMERS\", \"RECORD_IDS\": [ { \"ID\":
    \"2\" }, { \"ID\": \"3\" } ], \"RECORD_DICTCLAUSE\": \"\" } }");
    curl_setopt($curl, CURLOPT_HTTPHEADER, array('Content-Type: application/json'));

    curl_setopt($curl, CURLOPT_URL, "http://yourdomain.com:11201/api/read");
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    $result = curl_exec($curl);
    curl_close($curl);

} catch (Exception $e) {
    $result = $e->getMessage();
}
```

---

```
try
{
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS,
    "<LkCommand><APIKey>my_apikey</APIKey><Data><CALCULATED>False</CALCULATED><DICTIONARIES>False</DICTIONARIES><CONVERSION>False</CONVERSION><FORMAT_SPEC>False</FORMAT_SPEC><ORIGINAL_RECORDS>False</ORIGINAL_RECORDS><CUSTOM_VARS></CUSTOM_VARS><OUTPUT_FORMAT>MV</OUTPUT_FORMAT><FILE_NAME>LK.CUSTOMERS</FILE_NAME><RECORD_IDS><ID>2</ID><ID>3</ID></RECORD_IDS><RECORD_DICTCLAUSE></RECORD_DICTCLAUSE></Data></LkCommand>");
    curl_setopt($curl, CURLOPT_HTTPHEADER, array('Content-Type: text/xml'));

    curl_setopt($curl, CURLOPT_URL, "http://yourdomain.com:11201/api/read");
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    $result = curl_exec($curl);
    curl_close($curl);

} catch (Exception $e) {
    $result = $e->getMessage();
}
```

# LINKAR REST API con JavaScript, .NET, JAVA, PHP, cURL

## Ejemplo de Read con cURL

```
"curl -d '{ \"APIKey\" : \"xxxxxxx\", \"Data\" : { \"FILE_NAME\" : \"LK.CUSTOMERS\"  
, \"RECORD_IDS\" : [ { \"ID\" : \"2\" } ] } }' -H \"Content-Type: application/json\"  
http://linkerserverip:linkarmanagerport/api/read"
```

**LINKAR REST API con JavaScript, .NET, JAVA,  
PHP, cURL**

**GRACIAS**

**[www.kosday.com](http://www.kosday.com)**

**support@kosday.com**