

Guía Técnica para el Uso de la API de Mercado Público de Chile

Introducción

La **API de Mercado Público de Chile** (ChileCompra) es un servicio web que permite acceder de forma programática a la información de las compras públicas de Chile. A través de esta API es posible **consultar licitaciones (concursos públicos)** y **órdenes de compra** generadas en la plataforma MercadoPublico.cl, donde más de 850 organismos del Estado chileno compran bienes y servicios a miles de proveedores ¹. ² La API está en formato **REST** y soporta respuestas en **JSON**, **XML** o **JSONP**, brindando datos detallados de licitaciones (como su estado, comprador, adjudicatarios, etc.) y órdenes de compra (como monto, proveedor, estado de recepción, etc.).

Nota: La API actualmente tiene un estado **beta**, por lo que puede experimentar interrupciones ocasionales ³ ⁴. Además, está enfocada en **consulta de datos públicos**; no existen métodos documentados para realizar acciones transaccionales como enviar ofertas o modificar datos mediante la API pública.

Acceso a la API (Ticket de Autenticación)

Para utilizar la API se requiere un **Ticket de Acceso (API Key)** personal. Este ticket se obtiene registrándose en el portal de la API Mercado Público, sección "Participa", usando la credencial única del Registro Civil (ClaveÚnica) ⁵. Tras aceptar términos de uso y completar la solicitud ("Solicitud de Ticket"), el sistema enviará el código de acceso por correo electrónico (el proceso puede tomar tiempo según experiencias de usuarios).

- **Ticket de prueba:** La plataforma provee un ticket genérico para realizar pruebas iniciales sin registro. El ticket de prueba es `F8537A18-6766-4DEF-9E59-426B4FEE2844` ⁶. Este ticket público permite probar la API con datos de ejemplo, pero para un uso definitivo se debe obtener un ticket propio asociado a su cuenta.
- **Uso seguro del ticket:** Trate su ticket de acceso como una contraseña. Use siempre **HTTPS** para las consultas (la URL oficial ya lo implementa) y **no exponga** el ticket en código fuente público ni lo comparta. Cada solicitud debe incluir el ticket como parámetro GET (`&ticket=<TU_TICKET>`). Sin un ticket válido, la API no retornará datos.

Estructura General de la API

La API responde a **peticiones GET** sobre URLs bien definidas, las cuales incluyen los parámetros que determinan la consulta. La estructura base de las URLs es:

```
https://api.mercadopublico.cl/servicios/v1/publico/<Recurso>.<formato>?<parametros>&ticket=<TU_TICKET>
```

Donde `<Recurso>` puede ser, por ejemplo, **licitaciones** u **ordenesdecompra**, y `<formato>` puede ser `json`, `xml` o `jsonp`. Si no se especifica un formato, la respuesta por defecto será en **JSON** ⁷. A continuación, se describen los recursos principales y sus parámetros:

- **Licitaciones:** Corresponde a las distintas **convocatorias públicas** (concursos) publicadas en el portal. El recurso se denomina `licitaciones` y permite filtrar por código de licitación, fecha, estado, organismo comprador o proveedor, según se detalla más adelante.
- **Órdenes de Compra:** Representan las órdenes generadas a partir de licitaciones adjudicadas (o compras directas). El recurso se denomina `ordenesdecompra` y permite filtrar por código de O.C., fecha, estado, organismo o proveedor.
- **Otros recursos de utilidad:** La API también ofrece métodos para obtener ciertos códigos internos, por ejemplo: `BuscarProveedor` y `BuscarComprador` para obtener el código interno de un proveedor u organismo a partir de su RUT, tal como se explica más abajo en esta guía.

Formato de fechas: Muchos servicios requieren un parámetro de fecha para filtrar datos por día. El formato de fecha usado es **ddMMyyyy** (día, mes, año con cero inicial) ⁸ ⁹. Por ejemplo, `fecha=12062014` refiere al 12 de junio de 2014. Cuando se consulta **sin indicar fecha**, se asume el día actual (hoy), y cuando se consulta sin indicar estado, se incluyen *todos los estados* por defecto.

A continuación, profundizaremos en las funcionalidades específicas de la API, con ejemplos en **Python** para ilustrar su uso y mostrar evidencias de las respuestas obtenidas.

Consulta de Licitaciones (Búsqueda de concursos públicos)

La API permite realizar **búsquedas de licitaciones** usando diferentes criterios. Todas estas consultas se realizan contra la URL base `.../licitaciones.<formato>?...` incluyendo siempre el `ticket`. Los parámetros más comunes son:

- `codigo`: Permite obtener **una licitación específica** mediante su código único (ID de licitación en Mercado Público).
- `fecha`: Permite obtener las licitaciones **publicadas en una fecha dada** (formato ddMMyyyy).
- `estado`: Permite filtrar licitaciones por su **estado** (publicada, cerrada, adjudicada, etc.). Si se combina con `fecha`, trae las de cierto estado en esa fecha; si se usa `estado=activas` sin fecha, devuelve las licitaciones vigentes al día de hoy ¹⁰.
- `CodigoOrganismo` y `CodigoProveedor`: Permiten filtrar por el código interno de un **organismo público** (comprador) o de un **proveedor**. Generalmente se usan en conjunto con `fecha` para acotar la búsqueda a las licitaciones de ese día asociadas a dicho organismo o proveedor ¹¹ ¹².

A continuación se detallan los tipos de consulta disponibles para licitaciones, con ejemplos:

Búsqueda por código de licitación (detalle de una licitación específica)

Si se conoce el código único de una licitación (por ejemplo, "1509-5-L114"), se puede obtener **el detalle completo** de esa licitación. Esta consulta ignora cualquier parámetro de fecha, ya que devuelve la licitación solicitada sin importar cuándo ocurrió ¹³. Incluye todos los campos disponibles: datos del comprador, descripción, ítems o productos solicitados, participantes, adjudicaciones, etc.

Ejemplo: Obtener la licitación con código 1509-5-L114 en formato JSON.

```
import requests

ticket = "F8537A18-6766-4DEF-9E59-426B4FEE2844" # Ticket de prueba
codigo = "1509-5-L114"
url = f"https://api.mercadopublico.cl/servicios/v1/publico/licitaciones.json?
codigo={codigo}&ticket={ticket}"
respuesta = requests.get(url)
datos = respuesta.json()
print("Nombre licitación:", datos["Listado"][0]["Nombre"])
print("Estado:", datos["Listado"][0]["Estado"])
```

Evidencia de respuesta: La respuesta JSON contendrá la estructura general con campos como Cantidad (cantidad de licitaciones consultadas, que debería ser 1), FechaCreacion (fecha de la consulta) y un arreglo Listado con la información de la licitación. Un ejemplo simplificado de la estructura de respuesta sería:

```
{
  "Cantidad": 1,
  "FechaCreacion": "/Date(1696370400000-0300)/",
  "Version": "1.3.0",
  "Listado": [
    {
      "CodigoExterno": "1509-5-L114",
      "Nombre": "Adquisición de Equipos de Computación",
      "CodigoEstado": 8,
      "Estado": "Adjudicada",
      "FechaCierre": "/Date(1692211200000-0400)/",
      "Descripcion": "Compra de computadores para oficinas regionales...",
      "Comprador": {
        "NombreOrganismo": "Ministerio de Educación",
        "RutUnidad": "12345678-9",
        "...": "..."
      },
    },
    "Items": {
      "Cantidad": 3,
      "Listado": [
```

```

    {
      "NombreProducto": "Computador de escritorio",
      "Cantidad": 10,
      "Adjudicacion": {
        "RutProveedor": "76543210-1",
        "NombreProveedor": "Tecnología XYZ Ltda.",
        "MontoUnitario": 500000
      }
    },
    { "... item 2 ..."},
    { "... item 3 ..."}
  ]
}
// ...otros campos de la licitación...
}
]
}

```

(Los datos mostrados arriba son ilustrativos; en una respuesta real los valores dependerán de la licitación consultada.)

En este ejemplo, podemos observar que la licitación fue **Adjudicada** (CodigoEstado = 8 corresponde a Adjudicada ¹⁴) y se listan detalles del proveedor adjudicado dentro de cada ítem. Para cada licitación individual, la API entrega información exhaustiva, facilitando la extracción de todos los datos relevantes de ese proceso de compra.

Listar licitaciones diarias (todas las licitaciones de un día)

Para obtener un **listado de licitaciones** publicadas en una fecha específica, se utiliza el parámetro `fecha`. Esta consulta devuelve **todas las licitaciones publicadas ese día, en todos sus estados**, o bien se puede combinar con un filtro de estado específico (ver siguiente punto). La respuesta lista múltiples licitaciones con **información básica** de cada una (por ejemplo: código, nombre, estado, organismo) ¹⁵, a diferencia de la búsqueda por código que entrega información completa.

- Si se omite el parámetro `fecha`, la API asume la fecha actual. Es decir, `.../licitaciones.json?ticket=X` retornará las licitaciones del día en curso ¹⁶.
- El campo `Cantidad` en la respuesta indica cuántas licitaciones fueron encontradas para ese día.

Ejemplo: Listar todas las licitaciones **publicadas el 02/02/2014** (usando el ticket de prueba). Nótese el formato de la fecha `ddmmaaaa` en la URL.

```

fecha = "02022014" # 02 de febrero de 2014
url = f"https://api.mercadopublico.cl/servicios/v1/publico/licitaciones.json?
fecha={fecha}&ticket={ticket}"
respuesta = requests.get(url)
datos = respuesta.json()

```

```
print("Licitaciones encontradas:", datos["Cantidad"])
for lic in datos["Listado"][:5]:
    # Imprimir las primeras 5 licitaciones con su nombre y estado
    print(f"- {lic['CodigoExterno']}: {lic['Nombre']} (Estado: {lic['Estado']})")
```

Este código imprimiría la cantidad total de licitaciones encontradas ese día y listaría algunas con su nombre y estado. La estructura JSON incluirá un arreglo `Listado` donde cada elemento tiene campos como `CodigoExterno`, `Nombre`, `CodigoEstado`, `Estado`, `Comprador` (con subcampos como `NombreOrganismo`), etc. Dado que esta es una búsqueda masiva, **no se incluyen los detalles de items ni adjudicaciones** en la respuesta para cada licitación (para obtener detalles completos es necesario consultar cada código individualmente si se requiere).

Búsqueda de licitaciones activas (publicadas y no cerradas)

La API ofrece un atajo útil para obtener las licitaciones *abiertas* al momento actual. Usando `estado=activas`, se listan todas las licitaciones que están **publicadas y vigentes** (no han cerrado) en el día de la consulta ¹⁰. Equivale a obtener las licitaciones con estado "Publicada" del día presente.

Ejemplo: Obtener las licitaciones activas de hoy:

```
url = f"https://api.mercadopublico.cl/servicios/v1/publico/licitaciones.json?estado=activas&ticket={ticket}"
datos = requests.get(url).json()
print("Licitaciones activas hoy:", datos["Cantidad"])
primera = datos["Listado"][0]
print("Ejemplo:", primera["CodigoExterno"], "-", primera["Nombre"])
```

La consulta anterior retornará las licitaciones vigentes hoy, y en el ejemplo mostramos el código y nombre de la primera licitación activa en la lista. Internamente, `estado=activas` es un alias que filtra las licitaciones con estado *Publicada* a la fecha actual, facilitando la búsqueda de oportunidades abiertas sin tener que conocer la fecha exacta ni el estado numérico.

Según la documentación, el parámetro `estado` también puede tomar otros valores de estado textual: `"publicada"`, `"cerrada"`, `"desierta"`, `"adjudicada"`, `"revocada"`, `"suspendida"` o `"todos"` (para obtener todos los estados) ¹⁷. Usar `estado=publicada` en la fecha actual sería equivalente a `estado=activas`. Estos filtros permiten refinar la búsqueda diaria si solo interesan ciertos estados.

Búsqueda por estado específico en una fecha

Como combinación de los dos anteriores, es posible consultar **las licitaciones de un determinado estado en una fecha dada**. Por ejemplo, todas las licitaciones **adjudicadas el 02/02/2014**. Para ello se usan conjuntamente `fecha` y `estado`.

Ejemplo: Licitaciones **adjudicadas** el 02 de febrero de 2014.

```
GET /servicios/v1/publico/licitaciones.json?
fecha=02022014&estado=adjudicada&ticket=<TU_TICKET>
```

Según la documentación, este tipo de consulta devolverá las licitaciones de ese día en el estado indicado (en el ejemplo, estado "8" que corresponde a *Adjudicada*), mostrando información básica de cada una ¹⁸. De nuevo, la respuesta incluirá potencialmente múltiples resultados en **Listado**. Si la combinación de fecha y estado no encuentra registros, el campo **Cantidad** sería 0 y **Listado** vendría vacío.

Búsqueda por Organismo comprador o por Proveedor

La API también soporta filtrar las licitaciones **por organismo público** (unidad compradora) o por **proveedor** involucrado. Para utilizarlos, se debe conocer el **código interno** del organismo o proveedor en la plataforma Mercado Público. Estos códigos son identificadores numéricos propios del sistema (distintos del RUT). Por ejemplo, el código de organismo "Dirección de Compras y Contratación Pública" es 6945 ¹⁹, y el de un proveedor como la "Cámara de Comercio de Santiago" es 17793 ²⁰.

- **CodigoOrganismo=<ID>**: lista las licitaciones del día (o de una fecha específica si se combina con **fecha**) **convocadas por** el organismo con ese código ¹¹. Útil para monitorear lo licitado por una institución en particular.
- **CodigoProveedor=<ID>**: lista las licitaciones del día (o de una fecha) **donde participa** el proveedor con ese código ²¹. Esto usualmente se refiere a licitaciones en las que ese proveedor ha sido adjudicado u ofertado, según cómo exponga la API la información.

Ejemplo: Licitaciones publicadas el 02/02/2014 por el organismo con código 6945 (Dirección ChileCompra):

```
GET /servicios/v1/publico/licitaciones.json?
fecha=02022014&CodigoOrganismo=6945&ticket=<TU_TICKET>
```

Ejemplo: Licitaciones del 02/02/2014 donde participa el proveedor código 17793 (CCS):

```
GET /servicios/v1/publico/licitaciones.json?
fecha=02022014&CodigoProveedor=17793&ticket=<TU_TICKET>
```

Ambas consultas devolverán listas de licitaciones con información resumida. Es importante recalcar que para obtener estos códigos (organismo/proveedor) puede usarse la propia API, como se explica en la sección [Obtención de Códigos de Organismo y Proveedor](#).

En resumen, la API de licitaciones ofrece gran flexibilidad para buscar concursos públicos: desde obtener todos los procesos abiertos actualmente, hasta enfocarse en un número específico de licitación o en la actividad de una entidad o empresa particular en una fecha. Los datos básicos retornados (en las búsquedas por fecha/estado) incluyen típicamente el nombre de la licitación, organismo comprador, estado

y fechas relevantes como fecha de cierre. Para ahondar en los detalles de una licitación (bases, adjudicatarios, etc.), se usa la búsqueda por código individual.

Consulta de Órdenes de Compra (seguimiento de compras)

El segundo gran componente de la API son las **Órdenes de Compra (O.C.)**, que corresponden a los documentos generados cuando un organismo estatal concreta una compra con un proveedor a través del sistema. La API de órdenes de compra permite, de forma similar a licitaciones, consultar el estado y detalles de las O.C. utilizando distintos criterios:

- `codigo`: Obtener **una orden de compra específica** mediante su código único (ID de O.C. en Mercado Público).
- `fecha`: Listar las órdenes de compra **emitidas en una fecha** determinada.
- `estado`: Filtrar por estado de la orden (enviada, aceptada, cancelada, recepcionada, etc.).
- `CodigoOrganismo` / `CodigoProveedor`: Filtrar las órdenes por organismo comprador o por proveedor (usualmente proveedor adjudicado), en combinación con una fecha.

Los estados de las órdenes de compra se indican por nombre o código. Principales estados y sus códigos asociados incluyen: *Enviada al Proveedor (4)*, *En Proceso (5)*, *Aceptada (6)*, *Cancelada (9)*, *Recepción Conforme (12)*, *Pendiente de Recepcionar (13)*, *Recepcionada Parcialmente (14)* y *Recepción Conforme Incompleta (15)* ²². Estos estados se pueden usar como valor del parámetro `estado` en texto (ej. `estado=aceptada`). A continuación revisamos las consultas disponibles:

Consulta por código de Orden de Compra (detalle de una O.C.)

Si se cuenta con el identificador específico de una orden de compra (por ejemplo, "2097-241-SE14"), la API devuelve **todos los detalles de esa orden**. Al igual que con licitaciones, este método no depende de un parámetro de fecha; siempre retornará la orden correspondiente al código dado (si existe) ²³. La información detallada incluye el nombre de la orden (generalmente igual al de la licitación o compra), fechas (envío, aceptación, etc.), el organismo comprador, los datos del proveedor, los montos (neto, impuestos, total) y el desglose de ítems adquiridos.

Ejemplo: Obtener en JSON la orden de compra con código `2097-241-SE14`:

```
codigo_oc = "2097-241-SE14"
url = f"https://api.mercadopublico.cl/servicios/v1/publico/ordenesdecompra.json?
codigo={codigo_oc}&ticket={ticket}"
resp = requests.get(url)
orden = resp.json()["Listado"][0] # asumiendo que encontró la O.C.
print("Orden:", orden["Nombre"])
print("Estado proveedor:", orden["EstadoProveedor"])
print("Monto total:", orden["Total"])
```

Evidencia de respuesta: La estructura JSON de una orden de compra es similar a la de licitación en cuanto a campos globales (`Cantidad`, `FechaCreacion`, `Version`, etc.), pero dentro de `Listado` cada orden tiene múltiples subestructuras. Un ejemplo resumido:

```

{
  "Cantidad": 1,
  "FechaCreacion": "/Date(1696377600000-0300)/",
  "Version": "1.3.0",
  "Listado": [
    {
      "Codigo": "2097-241-SE14",
      "Nombre": "Compra de computadores para oficinas regionales",
      "CodigoEstado": 6,
      "Descripcion": "Orden de compra generada por adjudicación de licitación
1509-5-L114...",
      "TipoMoneda": "CLP",
      "TotalNeto": 5000000.0,
      "Impuestos": 950000.0,
      "Total": 5950000.0,
      "Comprador": {
        "CodigoOrganismo": 6945,
        "NombreOrganismo": "Dirección ChileCompra",
        "NombreUnidad": "Depto. de Informática",
        "NombreContacto": "Juan Pérez",
        "MailContacto": "jperez@chilecompra.cl"
      },
      "Proveedor": {
        "Codigo": 17793,
        "Nombre": "Tecnología XYZ Ltda.",
        "RutSucursal": "76543210-1",
        "NombreContacto": "María González",
        "MailContacto": "mgonzalez@xyz.cl"
      },
      "Items": {
        "Cantidad": 1,
        "Listado": [
          {
            "Correlativo": 1,
            "NombreProducto": "Computador de escritorio",
            "Cantidad": 10,
            "PrecioNeto": 500000.0,
            "Total": 5000000.0
          }
        ]
      },
      "Fechas": {
        "FechaEnvio": "/Date(1696380000000-0300)/",
        "FechaAceptacion": "/Date(1696466400000-0300)/",
        "FechaUltimaModificacion": "/Date(1696466400000-0300)/"
      },
      "EstadoProveedor": "Aceptada"
    }
  ]
}

```



```
}
]
}
```

(Nuevamente, los datos mostrados son representativos. En una respuesta real, los valores exactos dependerán del código consultado.)

Como se observa, la orden de compra incluye tanto los datos de la institución **Compradora** como del **Proveedor**. En este ejemplo ficticio, la orden está en estado "Aceptada" (lo que coincide con `CodigoEstado: 6` según la lista de códigos) y muestra un ítem comprado (10 computadores) con su precio y total. También se aprecian las fechas clave, como la de envío de la orden al proveedor y la fecha de aceptación por parte del proveedor.

Listar órdenes de compra diarias (todas las O.C. de un día)

Para obtener **todas las órdenes de compra emitidas en un día específico**, se utiliza el parámetro `fecha`. Por ejemplo, `fecha=02022014` retornará las órdenes emitidas el 2 de febrero de 2014 ²⁴. Si no se indica una fecha, por defecto se obtienen las órdenes **del día actual** (hoy). Esta consulta entrega un listado potencialmente grande de órdenes, cada una con **información básica**: código, nombre, estado, montos, etc., pero **no incluye el detalle completo de ítems** por eficiencia (similar al caso de licitaciones).

También es posible solicitar "todas las órdenes de compra del día actual" usando `estado=todos` explícitamente ²⁵, aunque este es el comportamiento por defecto al no filtrar estado.

Ejemplo: Obtener todas las órdenes de compra de hoy:

```
GET /servicios/v1/publico/ordenesdecompra.json?ticket=<TU_TICKET>
```

La respuesta listará todas las órdenes generadas en la fecha actual (si es día inhábil o no hubo actividad, `Cantidad` será 0). Cada elemento en `Listado` contendrá campos como `Codigo`, `Nombre`, `CodigoEstado`, `Total`, `Comprador` (nombre organismo) y `Proveedor` (nombre empresa), entre otros. Para mantener eficiencia, no se listan todos los ítems ni detalles de contacto en esta modalidad; para más detalles se debe consultar orden por orden con su código si se requiere profundizar.

Filtrar órdenes de compra por estado

Al igual que con licitaciones, es posible filtrar las órdenes de un día por su **estado**. Por ejemplo, obtener solo las órdenes *Aceptadas* hoy, o las *Canceladas* en cierta fecha. Los valores posibles para el parámetro `estado` en órdenes de compra son: `"enviadaproveedor"`, `"enproceso"` (aunque en la lista oficial el código 5 es "En proceso", la documentación muestra quizá un valor mal escrito; se esperaría `"enproceso"`), `"aceptada"`, `"cancelada"`, `"recepcionconforme"`, `"pendienterecepcion"` (pendiente de recepcionar), `"recepcionparcial"` (recepcionada parcialmente) y `"recepcionconformeincompleta"`, además de `"todos"` ²⁶. Estos corresponden a los estados 4,5,6,9,12,13,14,15 respectivamente.

Ejemplo: Órdenes **canceladas** en el día 02/02/2014:

```
GET /servicios/v1/publico/ordenesdecompra.json?
fecha=02022014&estado=cancelada&ticket=<TU_TICKET>
```

Esta consulta retornará únicamente las órdenes de compra del 2 de febrero de 2014 que hayan sido canceladas por el comprador o proveedor (estado código 9). La salida, de nuevo, será una lista filtrada con información básica de cada orden.

Usar estos filtros de estado puede ser útil para, por ejemplo, monitorear órdenes pendientes de recepción, o verificar cuáles órdenes fueron rechazadas/canceladas en cierto periodo, etc., sin tener que revisar manualmente cada orden.

Búsqueda de órdenes por Organismo o Proveedor

Analogamente a las licitaciones, podemos obtener órdenes de compra asociadas a un organismo público específico o a un proveedor específico:

- `CodigoOrganismo=<ID>`: Listará las órdenes de compra emitidas en la fecha dada **por** el organismo con ese código ²⁷. Sirve para revisar las compras realizadas por una institución en particular en cierto día.
- `CodigoProveedor=<ID>`: Listará las órdenes de compra emitidas en la fecha dada **hacia** el proveedor con ese código ²⁸. Es decir, compras donde ese proveedor fue el vendedor. Esto permite a un proveedor (o analista) ver las órdenes que un proveedor recibió en determinada fecha.

Estas consultas deben usarse generalmente con un parámetro `fecha` para acotar el resultado, ya que de lo contrario se interpretará por defecto el día actual. Por ejemplo:

- Órdenes del 02/02/2014 del organismo 6945 (ChileCompra):

```
.../ordenesdecompra.json?fecha=02022014&CodigoOrganismo=6945&ticket=...
```

- Órdenes del 02/02/2014 del proveedor 17793 (CCS):

```
.../ordenesdecompra.json?fecha=02022014&CodigoProveedor=17793&ticket=...
```

La respuesta en cada caso será la lista de órdenes que cumplen esa condición (cada orden con sus datos básicos). Si se dejara sin fecha, devolvería las del día actual para ese organismo/proveedor.

Finalmente, es importante mencionar que los **resultados de búsquedas de órdenes de compra** (cuando se listan múltiples) incluyen información básica suficiente para identificar la orden: por ejemplo, el estado en texto (EstadoProveedor), el total en pesos, y los nombres de comprador y proveedor ²⁹. Para profundizar detalles (ítems, fechas específicas), se recurriría a la consulta individual por código de O.C.

Obtención de Códigos de Organismo y Proveedor

Como vimos, algunas consultas requieren conocer el **Código interno** de un Organismo público o un Proveedor dentro del sistema Mercado Público. Estos códigos son distintos del RUT y actúan como identificadores en la API. La plataforma nos provee métodos para obtenerlos:

- **Buscar código de Proveedor por RUT:** Usando el recurso `Empresas/BuscarProveedor` se puede buscar el código de una empresa proveedora a partir de su RUT. Se debe proporcionar el RUT completo con formato (incluyendo puntos, guión y dígito verificador). Por ejemplo:

```
GET /servicios/v1/Publico/Empresas/BuscarProveedor?  
rutEmpresaProveedor=70.017.820-K&ticket=<TU_TICKET>
```

Este llamado devolverá el código interno correspondiente al RUT **70.017.820-K**, junto con el nombre de la empresa. Según la documentación, un ejemplo de resultado para ese RUT podría ser código **17793** y nombre "Cámara de Comercio de Santiago A.G. (CCS)" ²⁰. El campo `CodigoEmpresa` en la respuesta JSON corresponde al identificador del proveedor, y `NombreEmpresa` al nombre registrado ³⁰.

- **Listar códigos de Organismos Públicos:** Existe un método para obtener **todos los organismos públicos** registrados y sus códigos. El recurso es `Empresas/BuscarComprador` sin parámetros adicionales, solo con el ticket:

```
GET /servicios/v1/Publico/Empresas/BuscarComprador?ticket=<TU_TICKET>
```

Este método devuelve un listado de todos los organismos compradores del sistema con sus identificadores. Cada entrada tendrá un `CodigoEmpresa` (código del organismo) y `NombreEmpresa` (nombre de la institución) ³¹. Por ejemplo, en ese listado se encontraría que el código **6945** corresponde a "Dirección de Compras y Contratación Pública" ¹⁹, entre muchos otros.

Estos servicios son muy útiles para traducir datos conocidos (como el RUT de una empresa o el nombre de una institución) al código necesario para las consultas de licitaciones y órdenes. Se recomienda almacenar o cachear estos códigos si su aplicación va a consultar repetidamente la información de los mismos organismos o proveedores, para minimizar llamadas redundantes.

Envío de Ofertas a través de la API

Actualmente, **la API pública de Mercado Público** no permite el envío de ofertas **ni la realización de transacciones que modifiquen datos**; su alcance es principalmente de **lectura de información pública** (consulta de licitaciones y compras) ³. La presentación de ofertas en licitaciones debe realizarse a través del portal oficial (con login del proveedor y cumplimiento de los pasos en la plataforma web).

No se dispone de endpoints documentados para automatizar el ingreso de propuestas u ofertas económicas/técnicas mediante la API. Cualquier funcionalidad relacionada con ofertar electrónicamente estaría restringida a los sistemas oficiales y requeriría autenticación del usuario proveedor en el portal de MercadoPublico.cl, no habiendo un servicio REST público para ello.

En resumen: Si bien es posible usar la API para **obtener datos** (por ejemplo, listar licitaciones disponibles y luego decidir a cuál ofertar), el acto de **enviar la oferta** debe hacerse manualmente o mediante automatizaciones de interfaz (por ejemplo, *scripts* de navegación web) fuera del ámbito de la API. Siempre revise la documentación oficial o futuras actualizaciones en caso de que se habiliten nuevos servicios.

Buenas Prácticas y Consideraciones Finales

Al utilizar la API de Mercado Público, recomendamos seguir estas buenas prácticas para asegurar un uso eficiente y seguro:

- **Manejo seguro del Ticket:** Nunca incluya su ticket de acceso en código fuente público ni en repositorios sin seguridad. Trátelo como credencial sensible (por ejemplo, usar variables de entorno para cargarlo en su aplicación). Recuerde que todas las solicitudes requieren el ticket; si es comprometido, terceros podrían consumir la API en su nombre.
- **Uso de HTTPS y verificación de certificados:** La URL de la API es HTTPS por defecto; asegúrese de no degradar a HTTP. Esto garantiza que los datos (y el ticket) viajen cifrados. Verifique también los certificados en su cliente HTTP (las librerías usuales como `requests` de Python lo hacen por defecto).
- **Gestión de Errores:** Implemente manejo de errores en su código. La API podría retornar códigos HTTP de error o respuestas JSON con mensajes de error si algo falla (por ejemplo, ticket inválido, parámetros faltantes, servicio caído, etc.). Verifique el código de estado (`response.status_code`) y contemple reintentos con backoff exponencial si la API no responde, dado su estado beta. Si la respuesta JSON tiene `Cantidad=0` o campos vacíos, maneje esa situación en lugar de asumir que siempre habrá datos.
- **Formato de Fechas en JSON:** Note que las fechas en las respuestas JSON vienen en formato `/Date(timestamp)/`, por ejemplo `"/Date(1696370400000-0300)/"` para indicar un datetime en notación Unix Epoch con desfase de zona horaria. Para extraer un valor de fecha legible, deberá procesar esa cadena (por ejemplo, en Python se puede extraer el número entre paréntesis y convertirlo a datetime). Considere esta conversión al presentar fechas o compararlas.
- **Paginación o segmentación de consultas:** Si bien la API entrega datos por día (lo que limita naturalmente la cantidad de resultados por consulta), si necesita cubrir rangos amplios de tiempo, deberá iterar día por día o por rangos más pequeños. Evite hacer demasiadas llamadas en un corto intervalo de tiempo. Aunque no se especifican límites de tasa (rate limit) públicamente, un volumen muy alto podría saturar el servicio o causar bloqueos. Respete los recursos del API realizando consultas con intervalos razonables y, si es posible, programadas fuera de horarios pico.
- **Cacheo de resultados estáticos:** Información como la lista de organismos o incluso los detalles de una licitación adjudicada no cambiarán frecuentemente. Considere almacenar en caché (temporalmente) ciertas respuestas para evitar llamadas repetitivas innecesarias. Por ejemplo, si su aplicación consulta las mismas licitaciones varias veces al día, guardar esos datos durante unas horas puede reducir la carga.

- **Actualizaciones y cambios en la API:** Dada la naturaleza beta de la plataforma, es importante estar atento a anuncios oficiales. La documentación ha mostrado incorporaciones de nuevos servicios (como los de búsqueda por código interno en 2016) ³² ³³ . Revise periódicamente la sección de documentación de apoyo o comunicaciones de ChileCompra para adaptar su integración a posibles cambios o mejoras.
- **Seguridad y privacidad:** La data proporcionada por la API es pública (licitaciones y órdenes de compra son datos abiertos). Sin embargo, trate los datos con responsabilidad. Si cruza esta información con bases de datos internas, asegúrese de cumplir con la normativa de protección de datos personales cuando corresponda (por ejemplo, los datos de contacto de funcionarios o proveedores obtenidos a través de órdenes de compra). Asimismo, respete los términos de uso de la API ⁴ , evitando usos indebidos como *scraping* masivo fuera de los endpoints provistos o ataques de denegación de servicio.

En conclusión, la API de Mercado Público de Chile es una herramienta poderosa para acceder de forma automatizada a información de compras públicas. Permite a desarrolladores y analistas **buscar oportunidades de negocio (licitaciones abiertas)**, hacer seguimiento a **órdenes de compra** y obtener datos históricos, todo ello integrable en sistemas propios (por ejemplo, generar alertas, cuadros de mando, etc.). Con los ejemplos en Python presentados y las recomendaciones de uso, se puede comenzar a consumir esta API eficientemente, teniendo siempre en cuenta las limitaciones (solo lectura, estado beta) y responsabilidades asociadas. ¡Buenas consultas y análisis con la API de ChileCompra!

Fuentes consultadas: Documentación oficial de la API Mercado Público ¹⁵ ²⁹ , portal de información disponible de ChileCompra ⁶ ³⁴ , y materiales de referencia proporcionados. Todas las URLs y ejemplos mostrados han sido obtenidos o derivados de dichas fuentes oficiales de ChileCompra.

¹ Documentación API de Mercado Público - Órdenes de Compra.pdf

file:///file-3Q6Ns6pqTFJ7f4dBBHQxT4

² ⁷ Documentación API de Mercado Público - Órdenes de Compra | PDF

<https://es.scribd.com/document/698063465/Documentacion-API-de-Mercado-Publico-Ordenes-de-Compra>

³ ⁴ ³² ³³ API Mercado Público

<https://api.mercadopublico.cl/modules/DocuApoyo.aspx>

⁵ API Mercado Público

<https://api.mercadopublico.cl/modules/IniciarSesion.aspx>

⁶ ¹⁹ ²⁰ ³⁰ ³¹ ³⁴ Utilización - API Mercado Público

<https://api.mercadopublico.cl/modules/api.aspx>

⁸ ¹⁰ ¹¹ ¹² ¹³ ¹⁴ ¹⁵ ¹⁶ ¹⁷ ¹⁸ ²¹ API Mercado Público

https://api.mercadopublico.cl/modules/ejemplo_08.aspx

⁹ ²² ²³ ²⁴ ²⁵ ²⁶ ²⁷ ²⁸ ²⁹ API Mercado Público

https://api.mercadopublico.cl/modules/ejemplo_10.aspx