

# **GISA - Schnittstelle „Öffentliche Abfrage“**

Dokumentation des Webservices

Wien, 2022

## **Impressum**

Medieninhaber, Verleger und Herausgeber:

Bundesministerium für Arbeit und Wirtschaft, Stubenring 1, 1010 Wien  
Wien, 2022. Stand: 3. August 2022

### **Copyright und Haftung:**

Auszugsweiser Abdruck ist nur mit Quellenangabe gestattet, alle sonstigen Rechte sind ohne schriftliche Zustimmung des Medieninhabers unzulässig.

Es wird darauf verwiesen, dass alle Angaben in dieser Publikation trotz sorgfältiger Bearbeitung ohne Gewähr erfolgen und eine Haftung des Bundeskanzleramtes und der Autorin/des Autors ausgeschlossen ist. Rechtausführungen stellen die unverbindliche Meinung der Autorin/des Autors dar und können der Rechtssprechung der unabhängigen Gerichte keinesfalls vorgeifen.

Rückmeldungen: Ihre Überlegungen zu vorliegender Publikation übermitteln Sie bitte an [gewerbe@bmaw.gv.at](mailto:gewerbe@bmaw.gv.at).

# Inhalt

<b>1 Schnittstelle GISA - Öffentliche Abfrage</b> .....	<b>5</b>
1.1 Allgemeines .....	5
1.2 Versionen.....	6
1.3 Sicherheit.....	6
1.4 Endpunkt .....	6
1.4.1 SOAP Endpunkt.....	6
WSDL.....	6
Service URL.....	7
Gewerbe.....	7
GetGewerbe.....	7
1.4.2 REST Style Endpunkt im XML Format .....	8
GetGewerbe.....	8
Gewerbe.....	8
1.5 HTTP Status Codes.....	8
1.6 Beschreibung der Parameter.....	9
1.6.1 GewerbeRequest .....	9
1.6.2 GewerbeResponse.....	10
GewerbeResponse .....	10
Gewerbedaten .....	11
Berechtigung .....	12
Inhaber .....	14
NatPerson .....	14
JurPerson.....	15
Standort .....	16
1.7 Programmcode und Beispielaufufe .....	17
1.7.1 SOAP / Visual Studio - C#.....	17
Konfiguration .....	17
Serviceaufruf.....	17

1.7.2 REST/XML C#.....	18
1.7.3 Response-Beispiel (XML Format).....	19
<b>2 Versionshistorie .....</b>	<b>20</b>
Übersicht - Dokumentendaten .....	20

# 1 Schnittstelle GISA - Öffentliche Abfrage

## 1.1 Allgemeines

Die Schnittstelle enthält zwei Funktionen für die Abfrage der Gewerbedaten. Abfrageparameter sind jeweils GISA-Zahl und Name des Inhabers. Die beiden Funktionen liefern jeweils dasselbe Ergebnis und unterscheiden sich nur darin, wie die Abfrageparameter übergeben werden.

Die GISA-Zahl definiert das Gewerbe eindeutig – der Name des Inhabers dient als zusätzliches Kontrollkriterium. Als Name muss

- der Familienname für natürliche Personen oder
- der Firmenname für juristische Personen

übergeben werden.

Der Vergleich bezieht sich auf den Namen des Inhabers, berücksichtigt aber auch – wenn vorhanden – die Firma des Inhabers, den Fortbetrieb und den Pächter. Die Schreibweise des Namens muss nicht exakt übereinstimmen; der Vergleich erfolgt „tolerant“. Historische Namen werden ebenfalls berücksichtigt.

Ist das Gewerbe aufrecht und GISA-Zahl und Name stimmen überein, werden die Gewerbedaten wie unten beschrieben zurückgeliefert. Anderenfalls wird ein Fehlercode ungleich 0 und eine Fehlerbeschreibung geliefert.

Alle übermittelten Daten müssen UTF-8 codiert sein. Bei Verwendung anderer Zeichencodierungen kann das Service nicht erfolgreich aufgerufen werden. Die Ergebnisse werden ebenfalls UTF-8 codiert geliefert. UTF-8 ist als ISO-Standard veröffentlicht.

Übermittelte XML Objekte müssen valide sein.

## 1.2 Versionen

Derzeit ist die Version V1 der GISA Schnittstelle im Einsatz. Jede Änderung (neue oder geänderte Funktionen, Änderungen der Struktur der Ein- oder Ausgabeparameter) wird in einer neuen Version implementiert. Die alten Versionen bleiben dabei für maximal 1 Jahr unverändert in Betrieb. Die alten Versionen liefern `Deprecated true` im `GewerbeResponse` Status zurück. In diesem Fall sollte möglichst rasch eine Aktualisierung auf die neueste Version erfolgen. Die Versionen der Schnittstelle werden über eigene URLs adressiert.

## 1.3 Sicherheit

Der Zugriff erfolgt anonym. Das System limitiert die Anzahl der möglichen Requests pro IP- Adresse auf 100 pro Minute.

## 1.4 Endpunkt

Das Service bietet zwei Endpunkte für den Aufruf an. Die Endpunkte definieren die Art des Aufrufs und bestimmen das Format der Ein- und Ausgabewerte. Die nachfolgende detaillierte Beschreibung der Ein- und Ausgabeparameter erfolgt für das XML-Format, gilt aber auch für die SOAP Funktionen und Parameter.

### 1.4.1 SOAP Endpunkt

Bei SOAP handelt es sich um eine RPC Middleware die HTTP als Transportprotokoll und XML als Nachrichtenformat verwendet. Anders als beim nachfolgend beschriebenen Endpunkt im REST Style werden die Funktionen des SOAP Endpunktes mit derselben URL adressiert. Alle Aufrufe werden zunächst an den SOAP Dispatcher gerichtet. Dieser sorgt dann dafür, dass die in der Nachricht definierte Funktion aufgerufen wird.

### WSDL

Die WSDL ist die technische Schnittstellenbeschreibung für den SOAP Endpunkt und definiert auch die darin eingebundenen XML Schemadefinitionen. Über die WSDL kann die Schnittstelle automatisch eingebunden werden – Voraussetzung dafür ist ein

Werkzeug, welches die WSDL interpretiert und die Klassen für den Datenzugriff und die Datentransfer erzeugt (z.B. Visual Studio).

Die WSDL kann unter folgender URL geladen werden

- <https://www.gisa.gv.at/gisa-svc-public/GisaPublicV1.svc?singlewsdl>

### **Service URL**

Der SOAP Endpunkt ist über folgenden URL verfügbar:

- <https://www.gisa.gv.at/gisa-svc-public/GisaPublicV1.svc?singlewsdl>

Die generierte Service Zugriffsklasse (GisaPublicClient) implementiert folgende Funktionen:

### **Gewerbe**

Parameter: GewerbeRequest

Ergebnis: GewerbeResponse

Die Klassen GewerbeRequest, GewerbeResponse, GewerbeDto, wurden beim Einbinden des Services aus den Informationen der WSDL generiert. Vor dem Aufruf müssen die Abfragekriterien GISA-Zahl und Name des Inhabers den Eigenschaften des generierten GewerbeRequest Objektes zugewiesen werden.

Die Klassen und Eigenschaften entsprechen der nachfolgenden Beschreibung im XML Format.

### **GetGewerbe**

Parameter: GISAZahl und Name

Ergebnis: GewerbeResponse

Diese Funktion unterscheidet sich von der Funktion Gewerbe nur durch die Eingabeparameter.

Beispiele finden sich im Kapitel

## 1.4.2 REST Style Endpunkt im XML Format

Bei diesem Endpunkt wird kein Aufruf-Code generiert. Die Service Funktionen dieses Endpunktes werden jeweils über eine eigene URL aufgerufen.

Die REST xml Endpunkt wird über folgende URL (= <BaseURL>) aufgerufen:

- <https://www.gisa.gv.at/gisa-svc-public/GisaPublicV1.svc/xml>

Folgende Funktionen sind definiert:

### GetGewerbe

Hinweis: Die Funktion „GetGewerbe“ kann für einen Testaufruf im Browser nützlich sein. Sie darf jedoch nicht für eine produktive Schnittstellenanbindung verwendet werden, da es zu Problemen mit Sonderzeichen im Namen (als Bestandteil der URL) kommen wird. Es ist hier der Aufruf mit POST („Gewerbe“) zu verwenden.

URL: <BaseURL>/Gewerbe/{GISA-Zahl}/{Name}  
Methode: GET  
Ergebnis: GewerbeResponse - XML

### Gewerbe

URL: <BaseURL>/Gewerbe  
Methode: POST  
POST-Container: GewerbeRequest – XML  
Ergebnis: GewerbeResponse – XML

Beispiele finden sich im Kapitel 1.7.

## 1.5 HTTP Status Codes

Jeder Service Aufruf (Gewerbe bzw. GetGewerbe) antwortet mit einem standardisierten HTTP-Statuscode. Wurde die Anfrage erfolgreich bearbeitet, wird ein Statuscode der Klasse 2 (in der Regel der HTTP-Statuscode 200) zurückgeliefert.



Im Fehlerfall wird einer der definierten Fehler Statuscodes der Klasse 4 oder 5 zurückgeliefert.

- 4xx – Client-Fehler
- 5xx – Server-Fehler

Gründe für Fehler sind beispielsweise:

- die Service URL ist nicht korrekt
- das Service steht (temporär) nicht zur Verfügung
- das Service ist überlastet und kann den Request nicht annehmen
- die Eingabeparameter sind nicht korrekt
- ein interner Serverfehler

Bei erfolgreicher Bearbeitung der Anfrage (HTTP-Statuscode 2xx) liefern die Service Funktionen je nach Endpunkt ein GewerbeResponse-Objekt oder GewerbeResponse-XML zurück.

## 1.6 Beschreibung der Parameter

Im Folgenden werden alle Parameter im XML Format beschrieben. Die Beschreibung der Werte gilt auch für die generierten DTO Klassen beim SOAP Endpunkt.

### 1.6.1 GewerbeRequest

GewerbeRequest fasst die beiden erforderlichen Eingabeparameter GISA-Zahl und Name des Inhabers zusammen.

```
<GewerbeRequest>
  <GISAZahl>{GISA-Zahl}</GISAZahl>
  <Name>{Familiename oder Firma}</Name>
</GewerbeRequest>
```

Parameter/Element	Beschreibung	Anmerkung
<b>GISA-Zahl</b>	Maximal 10-stellige Ziffernfolge, die dauerhaft einer bestimmten Gewerbeberechtigung zugeordnet bleibt. GISA-Zahlen werden chronologisch vergeben.	Numerischer Wert
<b>Name</b>	Name der Person, welcher die Gewerbeberechtigung zugeordnet ist. Die bezeichnete Person ist der Gewerbeinhaber.	TD

### 1.6.2 GewerbeResponse

Beide Funktionen liefern für beide Endpunkte dasselbe Ergebnis. Die Formatierung ist vom Endpunkt abhängig.

Bei der Anbindung über SOAP wurde für GewerbeResponse eine entsprechende Klasse generiert. Das Ergebnis ist eine Instanz dieser Klasse. Die Eigenschaften dieser Klasse enthalten die Detailergebnisse.

Der XML Endpunkt liefert ein XML „Dokument“ mit den Ergebnisdaten.

#### GewerbeResponse

```

<GewerbeResponse>
  <Status>
    <Code>{Code}</Code>
    <Beschreibung>{Beschreibung}</Beschreibung>
    <Deprecated>{true/false}</Deprecated>
  </Status>
  <Gewerbe>
    .....
  </Gewerbe>
</GewerbeResponse>

```

Das Element „Status“ ist immer im GewerbeResponse enthalten und gesetzt. Das Element „Gewerbe“ wird nur dann geliefert, wenn der Code (im Element Status) gleich 0 ist.

Die folgende Tabelle beschreibt die GISA Public Status Codes:

Status Code	Beschreibung	Anmerkung
0	OK, Gewerbe wurde gefunden und zurückgeliefert	Gewerbe Element und Gewerbedaten sind vorhanden
1	Gewerbe nicht aufrecht	Gewerbe ist vorhanden aber nicht mehr aufrecht oder zur Gänze ruhend; Keine Gewerbedaten werden zurückgeliefert.
2	Gewerbe mit GISAZahl nicht vorhanden bzw. Name stimmt nicht überein	Kontrolle von GISAZahl und Name über Onlineabfrage erforderlich. Auch, wenn das Gewerbe noch niemals freigegeben wurde. Keine Gewerbedaten werden zurückgeliefert.

Deprecated ist *false*, wenn die aktuelle Version der Schnittstelle verwendet wird und *true*, wenn es eine neuere Version gibt.

## Gewerbedaten

```
...
<Gewerbe>
  <GISAZahl>{GISA-Zahl}</GISAZahl>
  <Berechtigung>
    ....
  </Berechtigung >
  <Inhaber>
    ....
  </Inhaber>
  <Standort>
    ....
  </Standort>
</Gewerbe>
```

## Berechtigung

```

...
<Berechtigung>
  <Kennzahl>{Kennzahl}</Kennzahl>
  <Wortlaut>{Wortlaut}</Wortlaut>
  <Ruhend>
    <Immobilienmakler>{true/false}</Immobilienmakler>
    <Immobilienverwalter>{true/false}</Immobilienverwalter>
    <Immobilienbautraeger>{true/false}</Immobilienbautraeger>
  </Ruhend>
  <Kreditvermittlung>
    <Code>{Kreditvermittlung-Code}</Code>
    <Beschreibung>{Beschreibungstext}</Beschreibung>
  </Kreditvermittlung>
  <RLBBerechtigt>{Reiseleistungsausübungsberechtigung true/false}</RLBBerechtigt>
</Berechtigung>

```

Parameter/Element	Beschreibung	Anmerkung
Kennzahl	Die Kennzahl ist eine höchstens sechsstellige Zahl, die den einzelnen Gewerbewortlauten zugeordnet ist („Gewerbeschlüssel“). Eine vollständige Tabelle dieser Kennzahlen steht aktualisiert zur Verfügung.	Numerischer Wert
Wortlaut	Der Wortlaut (allgemein auch als „Gewerbewortlaut“ bekannt) beschreibt den Umfang der Berechtigung, innerhalb welcher der Inhaber seine gewerblichen Tätigkeiten ausüben darf. Der Wortlaut ist somit eine generalisierte Beschreibung der gewerblichen Tätigkeiten, die ein Inhaber ausüben darf.	Wortfolge in lateinischen Buchstaben. Die Wortfolge kann auch Verweise auf gesetzliche Bestimmungen in Paragraphenform enthalten.
Ruhend	Wenn true: Schränkt den Gewerbewortlaut solange ein, als true gilt. Jede kann unabhängig von den anderen true oder false sein.	Besteht das Gewerbe aus mehreren Teilgewerben (Immobilienmakler, Immobilienvermittler, Bauträger), so kann in diesem Block die Information entnommen werden, ob eines der Teilgewerbe ruhend ist. Sind alle Teilgewerbe ruhend, so wird das ganze Gewerbe nicht ausgegeben (Status Code „1“)

Parameter/Element	Beschreibung	Anmerkung
Immobilienmakler	Wenn true: Immobilienmakler darf auch dann nicht ausgeübt werden, wenn der Wortlaut die Berechtigung zum Immobilienmakler einschließt.	Wenn false: Gewerbewortlaut bleibt unberührt.
Immobilienvermittler	Wenn true: Immobilienvermittler darf auch dann nicht ausgeübt werden, wenn der Wortlaut die Berechtigung zum Immobilienvermittler einschließt.	Wenn false: Gewerbewortlaut bleibt unberührt.
Immobilienbautraeger	Wenn true: Immobilienbauträger darf auch dann nicht ausgeübt werden, wenn der Wortlaut die Berechtigung zum Immobilienbauträger einschließt.	Wenn false: Gewerbewortlaut bleibt unberührt.
Kreditvermittlung	Zusatzinformation zum Gewerbewortlaut, die den Wortlaut weiter spezifiziert und in Einheit mit dem Wortlaut zu verstehen ist. Möglich ist nur einer der drei nachstehend beschriebenen Zustände.	Dieses Element existiert nur, wenn eine Kreditvermittlungsart definiert ist.
Code Beschreibung	<p>Gebunden: der Kreditvermittler kann nur auf einen Kreditgeber oder eine Gruppe von Kreditgebern zurückgreifen, die keine Marktmehrheit bilden. Solche Vermittler haben nicht das Recht, sich als „unabhängiger Kreditmakler“ zu bezeichnen.</p> <p>Ungebunden: Der Vermittler kann auf eine Vielfalt von Kreditgebern zurückgreifen. Solche Vermittler haben das Recht, sich als „unabhängiger Kreditmakler“ zu bezeichnen.</p> <p>Nicht berechtigt: schränkt den Gewerbewortlaut ein. Kredite dürfen auch dann nicht vermittelt werden, wenn der Wortlaut die Berechtigung zur Kreditvermittlung einschließt.</p>	<p>Kreditvermittlungsarten:</p> <ul style="list-style-type: none"> <li>• gebunden</li> <li>• ungebunden</li> <li>• nicht berechtigt</li> </ul>
RLBBerechtigung	<p>Wenn true: Es dürfen Pauschalreisen veranstaltet und verbundene Reiseleistungen zugesagt werden.</p> <p>Wenn false: Es dürfen keine Pauschalreisen veranstaltet und keine verbundene Reiseleistungen zugesagt werden.</p>	Wenn false: Wirkt wie eine Einschränkung des Gewerbewortlautes

#### Ruhend:

Das Element <Ruhend> wird nur dann ausgegeben, wenn für den Gewerbeschlüssel Teile des Gewerbes ruhend sein können. Die Elemente von <Ruhend> (Immobilienmakler, ...) sind ebenfalls optional und vom Gewerbeschlüssel abhängig.

## Inhaber

```
...
<Inhaber>
  <NatPerson>
    ....
  </NatPerson>
</Inhaber>
```

oder

```
...
<Inhaber>
  <JurPerson>
    ....
  </JurPerson>
</Inhaber>
```

Der Inhaber ist entweder eine natürliche oder juristische Person. Abhängig davon ist das Element <NatPerson> oder <JurPerson> vorhanden.

## NatPerson

```
...
<NatPerson>
  <Vorname>{Vorname}</Vorname>
  <Familiename>{Familiename}</Familiename>
  <Titel>{Titel}</Titel>
  <NachTitel>{nachgestellter Titel}</NachTitel>
  <GebDatum>{Geburtsdatum dd.mm.yyyy}</GebDatum>
</NatPerson>
```

Parameter/Element	Beschreibung
Vorname	Vorname einer natürlichen Person
Familiename	Nachname einer natürlichen Person
Titel	Dem Namen voranzustellende akademische Grade oder Standestitel (in der Regel: Doktorate, Magisterien, (Diplom)Ingenieure, Professuren und Berufsstandestitel)
NachTitel	Dem Namen hintanzustellende akademische Grade oder Standestitel (in der Regel: Abschlüsse von Post Graduate Ausbildungen)
GebDatum	Tag, an dem die natürliche Person geboren ist, ausgedrückt im gregorianischen Kalendersystem

## JurPerson

```
...
<JurPerson>
  <Firmenname>{Firmenname}</Firmenname>
  <FBNummer>{Firmenbuchnummer}</FBNummer>
  <ZVRZahl>{ZVRZahl}</ZVRZahl>
  <Rechtsform>
    <Code>{Rechtsformcode}</Code>
    <Bezeichnung>{Bezeichnung der Rechtsform}</Bezeichnung>
  </Rechtsform>
</JurPerson>
```

Nicht vorhandene Werte werden als leeres XML Element übergeben – z.B. <ZVRZahl/>

Parameter/Element	Beschreibung
Firmenname	Der Name, unter dem die juristische Person konstituiert ist
FBNummer	Wenn vorhanden: Zahl, unter der die juristische Person im österreichischen Firmenbuch registriert ist
ZVRZahl	Wenn vorhanden: Zahl, unter der die juristische Person im österreichischen Vereinsregister registriert ist
Rechtsform	Gibt Auskunft über die spezielle Rechtsform, unter welcher die juristische Person firmiert
Code	Rechtsform in Kurzbezeichnung, beispielsweise AG, GmbH, KG, OHG
Bezeichnung	Rechtsform in Langbezeichnung, beispielsweise: Aktiengesellschaft, Gesellschaft mit beschränkter Haftung, Kommanditgesellschaft, Offene Handelsgesellschaft

## Standort

```

...
<Standort>
  <Strasse>{Straße}</Strasse>
  <Hausnummer>{Hausnummer / Stiege / Stock / Tür}</Hausnummer>
  <PLZ>{Postleitzahl}</PLZ>
  <Ort>{Ortschaft}</Ort>
  <Gemeindenname>{Gemeinde}</Gemeindenname>
  <GKZ>{Gemeindekennzahl}</GKZ>
  <StaatCode>{Staat GWR Code}</StaatCode>
  <Staat>{Staatenname}</Staat>
</Standort>

```

Parameter/Element	Beschreibung	Anmerkung
Strasse	Name der Strasse	Kann mit dem Gemeindennamen identisch sein.
Hausnummer	Ordnungsnummer, welche für eine Gebäude oder eine Liegenschaft innerhalb des Straßenverlaufes vergeben wurde.	
PLZ	Zahl, die für ein bestimmtes Gebiet nach dem Postleitsystem von der österreichischen Post AG vergeben wurde.	Eine Postleitzahl kann sich auf unterschiedliche Gemeinden erstrecken.
Ort	Bezeichnung einer Ortschaft innerhalb einer Gemeinde. Ist keine Gebietskörperschaft sondern dient der exakten Beschreibung des Standortes in Gemeinden, die mehrere Ortschaften haben können. Insbesondere relevant, wenn ein Straßenname innerhalb einer Gemeinde mehrfach vergeben wurde.	Kann mit dem Gemeindennamen identisch sein.
Gemeindenname	Bezeichnung der Gebietskörperschaft Gemeinde.	
GKZ	Kennzahl der Gemeinde im Allgemeinen Gebäude- und Wohnungsregister.	Im Unterschied zum Postleitsystem jedenfalls eindeutige Zahlenkodierung einer Gemeinde
StaatCode	Kurzbezeichnung des Staates nach der Systematik des Allgemeinen Gebäude- und Wohnungsregisters.	
Staat	Langbezeichnung des Staates in deutscher Sprache.	Kann mit dem Gemeindennamen identisch sein.



## 1.7 Programmcode und Beispielaufufe

### 1.7.1 SOAP / Visual Studio - C#

Über die Funktion „Add Service Reference“ wird das SOAP Service zum Projekt hinzugefügt. Dabei ist die WSDL-URL für das Service einzugeben. Mit den Definitionen aus der WSDL werden die Zugriffs- und die Datenklassen generiert.

In der Konfigurationsdatei (web.config bzw. app.config) werden die Konfigurationselemente für das Service hinzugefügt. Diese Konfiguration muss ggf. vor dem Aufruf angepasst werden (Service-URL, HTTPS,...).

#### Konfiguration

```
<system.serviceModel>
  <bindings>
    <basicHttpBinding>
      <binding name="PublicSoapEndpoint">
        <security mode="Transport">
          <transport clientCredentialType="None"
            proxyCredentialType="None"></transport>
        </security>
      </binding>
    </basicHttpBinding>
  </bindings>
  <client>
    <endpoint address="<SERVICEURL>/soap"
      binding="basicHttpBinding"
      bindingConfiguration="PublicSoapEndpoint"
      contract="Gisa.Public.IGisaPublic" name="PublicSoapEndpoint"/>
  </client>
</system.serviceModel>
```

#### Serviceaufruf

```
try
{
    var request = new GewerbeRequest() { GISAZahl = 29111770, Name = "Test
Natperson 01" };
    var client = new GisaPublicClient();

    var res = client.Gewerbe(request);
    if (res.Status.Code == 0)
    {
        Console.WriteLine(res.Gewerbe.Berechtigung.Wortlaut);
        if (res.Gewerbe.Inhaber.NatPerson != null)
            Console.WriteLine("Inhaber {0} {1}",
                res.Gewerbe.Inhaber.NatPerson.Vorname,
                res.Gewerbe.Inhaber.NatPerson.Familienname);
        else if (res.Gewerbe.Inhaber.JurPerson != null)
            Console.WriteLine("Inhaber {0}",
                res.Gewerbe.Inhaber.JurPerson.Firmenname);
    }
}
```

```

res = client.GetGewerbe("29111770", "Test Natperson 01");
if (res.StatusCode == 0)
{
    ...
}
}
catch (Exception ex)
{
    ...
}

```

## 1.7.2 REST/XML C#

Das Rest-Service soll mittels Post die Parameter übergeben. Ein Beispiel:

```

POST https://www.gisa.gv.at/gisa-svc-public/GisaPublicV1.svc/xml/Gewerbe HTTP/1.1
Accept-Encoding: gzip,deflate
Content-Type: text/xml
Content-Length: 104
Host: www.gisa.gv.at
Connection: Keep-Alive
User-Agent: Apache-HttpClient/4.1.1 (java 1.5)

<GewerbeRequest>
  <GISAZahl>29111770</GISAZahl>
  <Name> Test Natperson 01</Name>
</GewerbeRequest>

```

Im .Net Framework gibt es Klassen für den Zugriff auf Webservice. Für den synchronen Zugriff beispielsweise die Klasse WebClient; für den asynchronen Zugriff die Klasse HttpClient. Ergebnis der Abfrage ist das GewerbeResponse Objekt als XML Dokument – siehe Beispiel in Kapitel 1.7.3

```

try
{
    string url = SERVICEURL + "/xml/Gewerbe";
    string xmlResult;
    string xmlRequest =
        @"<GewerbeRequest>
          <GISAZahl>29111770</GISAZahl>
          <Name> Test Natperson 01</Name>
        </GewerbeRequest>";

    using (var wc = new WebClient())
    {
        wc.Headers[HttpRequestHeader.ContentType] = "application/xml";
        byte[] requestData = Encoding.UTF8.GetBytes(xmlRequest);
        byte[] responseData = wc.UploadData(url, "POST", requestData);
        xmlResult = Encoding.UTF8.GetString(responseData);
    }

    XmlDocument doc = new XmlDocument();
    doc.LoadXml(xmlResult);
    XmlNode statusCode =
    doc.SelectSingleNode("/GewerbeResponse/Status/Code");
    if (statusCode.InnerText == "0")
    {
        ...
    }
}

```

```

}
catch (Exception ex)
{
    ...
}

```

### 1.7.3 Response-Beispiel (XML Format)

```

<?xml version="1.0"?>
<GewerbeResponse>
  <Status>
    <Code>0</Code>
    <Beschreibung>OK</Beschreibung>
    <Deprecated>>false</Deprecated>
  </Status>
  <Gewerbe>
    <GISAZahl>123456</GISAZahl>
    <Berechtigung>
      <Kennzahl>129901</Kennzahl>
      <Wortlaut>Elektromechaniker und Elektromaschinenbauer gem. &#xA7;
      94 &#x2026;</Wortlaut>
      <Ruhend>
        <Immobilienmakler>>false</Immobilienmakler>
        <Immobilienverwalter>>false</Immobilienverwalter>
        <Immobilienbautraeger>>false</Immobilienbautraeger>
      </Ruhend>
      <Kreditvermittlung>
        <Code>3</Code>
        <Beschreibung>nicht berechtigt</Beschreibung>
      </Kreditvermittlung>
      <RLBBerechtigt>>false</RLBBerechtigt>
    </Berechtigung>
    <Inhaber>
      <JurPerson>
        <Firmenname>Lux &#xD6;sterreich GmbH</Firmenname>
        <FBNummer>122944a</FBNummer>
        <ZVRZahl />
        <Rechtsform>
          <Code>GES</Code>
          <Bezeichnung>Gesellschaft mit beschr&#xE4;nkter
          Haftung</Bezeichnung>
        </Rechtsform>
      </JurPerson>
    </Inhaber>
    <Standort>
      <Strasse>Concorde Business Park</Strasse>
      <Hausnummer>1/B3/Top31</Hausnummer>
      <PLZ>2320</PLZ>
      <Ort>Rauchenwarth</Ort>
      <Gemeindename>Rauchenwarth</Gemeindename>
      <GKZ>30738</GKZ>
      <StaatCode>A</StaatCode>
      <Staat>&#xD6;sterreich</Staat>
    </Standort>
  </Gewerbe>
</GewerbeResponse>

```

## 2 Versionshistorie

### Übersicht - Dokumentendaten

<b>Titel</b>	GISA - Schnittstelle „Öffentliche Abfrage“
<b>Betreff</b>	GISA - Schnittstelle
<b>Verfasst von</b>	Magistrat Wien und Bundesministerium für Digitalisierung und Wirtschaftsstandort
<b>Vertraulichkeitsstufe</b>	Nicht eingeschränkt
<b>Gültigkeitszeitraum</b>	Unbegrenzt
<b>Freigabeinformation</b>	Freigabe am 30.11.2018 durch GISA-Support
<b>Version</b>	V1.0.0
<b>Ablageort</b>	<u><a href="https://bma.wg.at">GISA - Schnittstelle (bma.wg.at)</a></u>

V1.0.0 vom 29.11.2018  
unbeschränkt gültig  
Erstversion  
Freigabe am 30.11.2018 durch GISA-Support

**Bundesministerium für Arbeit und Wirtschaft**

Stubenring 1, 1010 Wien

+43 1 711 00-0

[gewerbe@bmaw.gv.at](mailto:gewerbe@bmaw.gv.at)

[bmaw.gv.at](http://bmaw.gv.at)