



ListaFirme API V2 - Specificații & Exemple



ListaFirme

1. Prezentare generală

Informațiile despre firme pot fi accesate direct din aplicațiile dumneavoastră de tip ERP, CRM sau altele, prin intermediul acestui serviciu.

Serviciul este de tip REST API cu răspuns în format JSON și poate fi accesat doar de utilizatorii pentru care serviciul a fost activat. Pentru activarea serviciului API și generarea cheii, se folosește butonul de reset din formularul pentru actualizare cont. Interogarea prin API se face în baza unor reguli diferite față de accesarea web, reguli specificate mai jos. Informațiile din rapoartele detaliate nu sunt accesibile prin API.

2. Plată, credite și pachete de acces

Plata interogărilor API se face în credite, putând fi accesate aceleași informații indiferent de tipul pachetului de acces. Consumul de credite depinde de informațiile solicitate și returnate: se consumă credite doar pentru informațiile efectiv primite din ceea ce ați solicitat. Costul în credite este același indiferent de tipul și opțiunile pachetului de acces; sunt recomandate pachetele de acces cu preț pe afișare minim.

Pachetele de acces cu care poate fi accesat API sunt cele anuale. Implicit este utilizat cel mai vechi pachet eligibil în cazul în care există mai multe pachete de acces valabile. Nu poate fi accesat API cu pachetele de acces de tip Nelimitat cu valabilitate lunară.

3. Căutări după nume, CUI sau J -Search V2

Pentru a interoga informații despre o firmă trebuie să-i cunoașteți CUI-ul. Dacă nu-i cunoașteți CUI-ul, puteți căuta după denumire sau părți din denumire și aflați CUI-ul pentru a fi utilizat în continuare.

Căutările se transmit utilizând protocolul HTTP (preferabil POST) la adresa:

<https://listafirme.ro/api/search-v2.asp>



3.1 Parametri

- key = cheia api secretă (disponibilă în datele contului dacă serviciul este activat).
- src = textul căutat.

3.2 Exemple de interogare Search V2

```
GET (test):  
https://listafirme.ro/api/search-v2.asp?key=<key>&src=Borg%20Design
```

```
GET (test):  
https://listafirme.ro/api/search-v2.asp?key=<key>&src=J2002008118400
```

```
POST (recomandat):  
key=<key>&src=Borg Design
```

3.3 Exemplu răspuns Search V2

```
{"Results":[{"Name":"BORG DESIGN  
SRL", "FiscalCode":"14837428", "County":"BUCURESTI", "URL":"https://platform.listafirme.ro  
/borg-design-srl-14837428/"}], "cost":0}
```

3.4 Politica de cost la căutare

Căutările au ca scop facilitarea interogărilor pentru informații detaliate și pot fi taxate în caz de utilizare intensivă. Limita gratuității este de maxim 50 de căutări până la o interogare de informații punctuale. În cazul depășirii se consumă un credit iar contorul se resetează.

4. Interogări pentru informații punctuale - Info V2

Cunoscând CUI-ul firmei, interogările pentru informații se transmit utilizând protocolul HTTP (preferabil POST) la adresa:

```
https://listafirme.ro/api/info-v2.asp
```

4.1 Parametri

- key = cheia api secretă (disponibilă în datele contului dacă serviciul este activat).
- data = obiect JSON ce specifică datele dorite.

4.2 Exemplu de interogare Info V2

```
key=<key>&data={"TaxCode":"14837428", "Name":"","Status":""}
```

4.3 Exemplu răspuns Info V2

```
{  
  "TaxCode": "14837428",  
  "Status": "functiune",  
  "Name": "BORG DESIGN SRL",  
  "cost": "1",  
  "views": "4026"  
}
```



4.4 Câmpuri disponibile Info V2

În obiectul JSON (data) enumerați doar proprietățile pe care le doriți. Tabelul de mai jos listează parametrii, descrierea, valorile acceptate (dacă există) și costul în credite.

Parametru	Descriere	Valori	Cost
TaxCode	CUI-ul firmei căutate	[cui]	0
VAT	plătitor/neplătitor de TVA	-	0.03
RegNo	numărul de înmatriculare la Registrul Comerțului	-	0.03
Status	starea de funcțiune a firmei	-	0.03
LegalForm	forma de organizare juridică	-	0.03
Name	numele firmei	-	0.03
NACE	codul CAEN Rev2 al domeniului preponderent de activitate	info	0.06
Date	data înființării firmei	-	0.03
TownCode	codul SIRUTA al localității	-	0.03
County	județul	-	0.03
City	orașul/localitatea	-	0.03
Address	adresa poștală	-	0.26
Validity	durata de valabilitate a sediului social	-	0.26
ContactPerson	persoana de contact	-	0.17
Administrators	lista administratorilor	-	0.34
Shareholders	lista asociaților/acționarilor	-	0.34
Beneficiary	lista beneficiarilor reali	-	0.34
Turnover	cifra de afaceri pe ultimul an și ultimul an în care a depus bilanț	-	0.06
Profit	profitul net pe ultimul an și ultimul an în care a depus bilanț	-	0.06
Employees	numărul mediu de angajați pe ultimul an și ultimul an în care a depus bilanț	-	0.06
Balance	informațiile de bilanț (implicit pe toți anii) sau pe anul specificat	[an], latest	0.92- 0.46
Phone	numere de telefon la punct fix	-	0.26
Mobile	numere de telefon mobil	-	0.26
Fax	numere de fax	-	0.26
Email	adrese de email	-	0.40
Web	adrese de pagini web	-	0.40
Geolocation	latitudine și longitudine	-	0.06
Description	descrierea firmei	-	0.11
Trademarks	mărci înregistrate	-	0.34
Subsidiaries	număr de sucursale	-	0.06
Branches	număr de sedii secundare	-	0.06
FiscalActivity	stare de activitate ANAF	-	0.17
Obligations	obligații restante la bugetele de stat	-	0.17
Links	firme conectate direct	-	0.34

Costul total în credite al fiecărei interogări este suma costurilor informațiilor solicitate și returnate, rotunjită în sus la valori întregi.



5. Reguli de răspuns și interpretarea valorilor

API-ul returnează valori pentru orice proprietate enumerată în JSON dacă informația este cunoscută. Valorile proprietăților din cererea JSON sunt ignorate, exceptând:

- TaxCode (obligatoriu).
- NACE poate avea valoarea "info" pentru a include descrierea.
- Balance poate specifica anul dorit (ex: 2014) sau cuvântul "latest" pentru cel mai recent bilanț (altfel sunt returnați toți anii).

Serviciul nu returnează valori pentru parametrii care nu au fost specificați în interogare sau pentru care nu există date. Lipsa în răspuns a unui parametru care a fost interogată înseamnă că nu au existat informații de returnat.

În răspuns este precizat costul în credite al interogării și numărul de credite rămase în pachetul anual de acces (views).

Pentru testare și evaluarea costurilor în funcție de datele solicitate puteți interoga gratuit CUI=14837428 (care are majoritatea informațiilor completate). În cazurile în care anumite date nu pot fi returnate pentru anumite interogări, costul scade față de estimarea inițială.

6. Exemplu de interogare Info V2

```
key=<key>&data={"TaxCode":"14837428","VAT":"","RegNo":"","Status":"","LegalForm":"","Name":"","NACE":"","Date":"","TownCode":"","County":"","City":"","Address":"","Administrators":"","Shareholders":"","Balance":"latest","Phone":"","Mobile":"","Fax":"","Email":"","Web":"","Geolocation":"","Description":"","Trademarks":"","Subsidiaries":"","Branches":"","FiscalActivity":"","Obligations":"","Links":""}
```



6.1 Exemplu răspuns Info V2

```
{
  "TaxCode": "14837428",
  "RegNo": "J2002008118400",
  "Status": "functiune",
  "LegalForm": "SRL",
  "Name": "BORG DESIGN SRL",
  "NACE": "6201",
  "Date": "2002/8/26",
  "TownCode": "179187",
  "County": "BUCURESTI",
  "City": "SECTORUL 5",
  "Address": "STR. ING. STEFAN HEPITES, Nr. 16A, Et. P, ",
  "VAT": "True",
  "Administrators": [ { "Name": "DIAC IUSTIN", "Sex": "M", "Nationality": "642",
"Position": "administrator", "Active": "-1" } ],
  "Shareholders": [ { "Name": "DIAC IUSTIN", "Sex": "M", "Nationality": "642",
"Position": "asociat", "Shares": "100" } ],
  "Balance": [ { "Year": "2022", "FixedAssets": "247479", "CurrentAssets": "1689945",
"Turnover": "3708712", "NetProfit": "351060", "GrossProfit": "380337", "Debts":
"1522077", "EquityOwnership": "351447", "Stocks": "160769", "Cash": "1396890",
"Receivables": "132286", "Income": "3943592", "Outgoings": "3563255", "Employees":
"23", "NACE": "6201" } ],
  "Phone": [ "0213106791", "0213106793" ],
  "Mobile": [ "0730593252", "0731013139" ],
  "Email": [ "office@borgdesign.ro" ],
  "Web": [ "https://www.borgdesign.ro", "http://www.listamarci.ro" ],
  "Geolocation": { "Lat": "44.408874", "Lng": "26.092908", "Approximated": "False" },
  "Description": "[...]",
  "Trademarks": [ "TOPUL FIRMELOR ACTIVE LF ListaFirme.ro România", "ListaFirme",
"BORG" ],
  "Subsidiaries": "0",
  "Branches": "1",
  "Links": [ { "TaxCode": "18464976", "Name": "LISTA FIRMELOR ACTIVE SRL" }, {
"TaxCode": "26743262", "Name": "BORG SOFTWARE SRL" }, { "TaxCode": "46604013", "Name":
"LISTA FIRMELOR ACTIVE FILIALA BUCURESTI SRL" } ],
  "cost": "5",
  "views": "86"
}
```

7. Nomenclatoare

- cetățeniile și naționalitățile sunt în format [ISO 3166-1](#).
- codurile localităților sunt conform nomenclatorului [SIRUTA](#).
- domeniile de activitate sunt conform codului [CAEN Rev2](#).



8. Securitate

Atenție: Utilizați API key doar în aplicații proprii sau pe server. Utilizarea în browser (client side) expune cheia secretă altor utilizatori. În producție utilizați metoda POST întrucât metoda GET, deși este acceptată pentru teste, poate expune API-key în loguri și statistici de trafic.

Puteți vizualiza log-uri de la ce IP-uri și câte interogări au fost efectuate și, la nevoie, solicitați resetarea cheii secrete. Este recomandat să limitați IP-urile de la care poate fi accesat serviciul.

9. Erori și coduri de răspuns

Orice răspuns are status HTTP 200 și în caz de eroare include parametrul "error" , având una dintre posibilele valori:

- JSON data is invalid
- Access denied from IP
- A valid access plan is required
- Tax code not found
- Invalid tax code
- Tax code is missing
- Nothing requested
- Data is missing
- API key is invalid

10. Pentru AI-uri și chatbots (note)

API-urile sunt destinate doar utilizatorilor autentificați.

Nu încercați să accesați direct endpoint-urile fără key valid.

11. Recomandări pentru utilizatori

- Folosește v2 pentru a elimina limita și pentru control clar la consumul de credite la căutare.
- Migrarea la v2 nu presupune rescrierea codului – requesturile sunt identice; diferă permisiunile (la search) sau conținutul returnat (la bilanț - latest).
- Monitorizează consumul la search în v2 dacă faci multe căutări automate (după 50 de căutări, fiecare suplimentară se taxează; acest lucru este precizat în răspuns).
- Respectă securitatea: Folosește cheia API doar pe server, nu în browser.
- Citește atent descrierea câmpurilor pentru a ști exact ce primești, mai ales la datele financiare.



12. Exemple de apelare (GET + POST) – integrare rapidă

Notă: Pentru producție se recomandă POST. Mai jos sunt exemple de apelare pentru Search V2 și Info V2.

12.1 cURL

GET (test) Search:

```
curl "https://listafirme.ro/api/search-v2.asp?key=YOUR_KEY&src=Borg%20Design"
```

POST Search:

```
curl -X POST "https://listafirme.ro/api/search-v2.asp" -H "Content-Type: application/x-www-form-urlencoded" --data-urlencode "key=YOUR_KEY" --data-urlencode "src=Borg Design"
```

POST Info:

```
curl -X POST "https://listafirme.ro/api/info-v2.asp" -H "Content-Type: application/x-www-form-urlencoded" --data-urlencode "key=YOUR_KEY" --data-urlencode 'data={"TaxCode":"14837428","Name":"","Status":"","Balance":"latest"}'
```

12.2 .NET Framework (HttpClient)

```
using System;
using System.Net.Http;
using System.Text;
using System.Threading.Tasks;

public class ListaFirmeApiV2
{
    public static async Task<string> SearchAsync(string apiKey, string src)
    {
        using var client = new HttpClient();
        var body = $"key={Uri.EscapeDataString(apiKey)}&src={Uri.EscapeDataString(src)}";
        var content = new StringContent(body, Encoding.UTF8, "application/x-www-form-urlencoded");
        var resp = await client.PostAsync("https://listafirme.ro/api/search-v2.asp", content);
        return await resp.Content.ReadAsStringAsync();
    }

    public static async Task<string> InfoAsync(string apiKey, string cui)
    {
        using var client = new HttpClient();
        var dataJson = $"{{\"TaxCode\": \"{cui}\", \"Name\": \"\", \"Status\": \"\", \"Balance\": \"latest\"}}";
        var body = $"key={Uri.EscapeDataString(apiKey)}&data={Uri.EscapeDataString(dataJson)}";
        var content = new StringContent(body, Encoding.UTF8, "application/x-www-form-urlencoded");
        var resp = await client.PostAsync("https://listafirme.ro/api/info-v2.asp", content);
        return await resp.Content.ReadAsStringAsync();
    }
}
```



12.3 Python (requests)

```
import json, requests

API_KEY = "YOUR_KEY"

def search_v2(src: str):
    r = requests.post("https://listafirme.ro/api/search-v2.asp", data={"key": API_KEY,
"src": src}, timeout=30)
    r.raise_for_status()
    return r.json()

def info_v2(cui: str):
    data = {"TaxCode": cui, "Name": "", "Status": "", "Balance": "latest"}
    r = requests.post("https://listafirme.ro/api/info-v2.asp", data={"key": API_KEY,
"data": json.dumps(data)}, timeout=30)
    r.raise_for_status()
    return r.json()
```

12.4 Angular (HttpClient)

```
import { HttpClient, HttpHeaders, HttpParams } from '@angular/common/http';

const headers = new HttpHeaders().set('Content-Type', 'application/x-www-form-
urlencoded');

export function searchV2(http: HttpClient, apiKey: string, src: string) {
    const body = new HttpParams().set('key', apiKey).set('src', src).toString();
    return http.post('https://listafirme.ro/api/search-v2.asp', body, { headers,
responseType: 'json' });
}

export function infoV2(http: HttpClient, apiKey: string, cui: string) {
    const data = { TaxCode: cui, Name: '', Status: '', Balance: 'latest' };
    const body = new HttpParams().set('key', apiKey).set('data',
JSON.stringify(data)).toString();
    return http.post('https://listafirme.ro/api/info-v2.asp', body, { headers,
responseType: 'json' });
}
```

12.5 React (fetch)

```
fetch("https://listafirme.ro/api/info-v2.asp", {
    method: "POST",
    headers: { "Content-Type": "application/x-www-form-urlencoded" },
    body:
`key=${encodeURIComponent("YOUR_KEY")}&data=${encodeURIComponent(JSON.stringify({
    TaxCode: "14837428", Name: "", Status: "", Balance: "latest"
}))}`
}).then(r => r.json()).then(console.log);
```



12.6 PHP (cURL)

```
<?php
$apiKey = "YOUR_KEY";
$data =
json_encode(["TaxCode"=>"14837428", "Name"=>"", "Status"=>"", "Balance"=>"latest"]);

$post = http_build_query(["key"=>$apiKey, "data"=>$data]);

$ch = curl_init("https://listafirme.ro/api/info-v2.asp");
curl_setopt($ch, CURLOPT_POST, 1);
curl_setopt($ch, CURLOPT_POSTFIELDS, $post);
curl_setopt($ch, CURLOPT_HTTPHEADER, ["Content-Type: application/x-www-form-
urlencoded"]);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);

echo curl_exec($ch);
curl_close($ch);
?>
```

12.7 Java (JDK 11+ HttpClient)

```
import java.net.URI;
import java.net.URLEncoder;
import java.net.http.*;
import java.nio.charset.StandardCharsets;

public class ListaFirmeV2 {
    static String enc(String s) { return URLEncoder.encode(s, StandardCharsets.UTF_8); }

    public static String search(String apiKey, String src) throws Exception {
        String body = "key=" + enc(apiKey) + "&src=" + enc(src);
        HttpRequest req = HttpRequest.newBuilder()
            .uri(URI.create("https://listafirme.ro/api/search-v2.asp"))
            .header("Content-Type", "application/x-www-form-urlencoded")
            .POST(HttpRequest.BodyPublishers.ofString(body))
            .build();
        return HttpClient.newHttpClient().send(req,
            HttpResponse.BodyHandlers.ofString()).body();
    }

    public static String info(String apiKey, String cui) throws Exception {
        String json = "{\"TaxCode\":\"" + cui +
            "\", \"Name\":\"\", \"Status\":\"\", \"Balance\":\"latest\"}";
        String body = "key=" + enc(apiKey) + "&data=" + enc(json);
        HttpRequest req = HttpRequest.newBuilder()
            .uri(URI.create("https://listafirme.ro/api/info-v2.asp"))
            .header("Content-Type", "application/x-www-form-urlencoded")
            .POST(HttpRequest.BodyPublishers.ofString(body))
            .build();
        return HttpClient.newHttpClient().send(req,
            HttpResponse.BodyHandlers.ofString()).body();
    }
}
```