FISCALNET

Manual tehnic de integrare

Windows, Android, fișiere de comandă, format INP, API HTTP și Android Intent

Document de referință pentru integratori, furnizori software și echipe de suport operațional

Acest manual descrie metodele oficiale de integrare, sintaxa comenzilor, modelele de răspuns și regulile de validare pentru fluxurile Windows și Android.

Generat la data: 30.04.2026 07:40

1. Scopul documentului și metodele de integrare

Acest manual are rolul de a documenta, într-o formă unitară, toate canalele prin care o aplicație externă poate transmite comenzi către FiscalNet și poate interpreta răspunsul rezultat.

Sunt acoperite patru scenarii principale: integrare prin fișier de comandă pe Windows, integrare prin API HTTP, integrare prin Android Intent și utilizarea formatului INP în aplicațiile care provin din ecosisteme mai vechi.

Metodă	Când este recomandată	Tipul de răspuns
Fișier de comandă	Aplicații desktop, ERP-uri vechi, automatizări locale și scenarii fără programare HTTP.	Fișier text de răspuns în folderul Raspuns.
API HTTP	Backend, middleware, servicii Windows, aplicații desktop moderne și integrare de rețea.	JSON sincron sau rezultat async prin job.
Android Intent	O altă aplicație Android instalată pe același terminal.	Rezultat întors prin Activity Result.
Format INP	Compatibilitate cu sisteme care generează deja fișiere INP.	Același mecanism de răspuns ca la fișierul de comandă.

Principiu de bază

Indiferent de canalul de integrare, aplicația execută aceeași logică funcțională. Se schimbă numai transportul: folder, HTTP sau Intent Android.

2. Convenții generale de sintaxă și validare

O comandă este alcătuită din una sau mai multe linii. Fiecare linie reprezintă o instrucțiune. În formatul text simplu separatorul de câmp este caracterul ^, iar în formatul INP separatorii sunt virgula și punctul și virgula.

Regulă	Explicație operațională
O instrucțiune pe linie	Fiecare linie transmite o singură operație. Ordinea liniilor este semnificativă.
Contextul se construiește progresiv	Unele comenzi, cum ar fi TL, DP, DV, MP și MV, depind de poziția lor în raport cu produsul curent sau subtotalul curent.
Sumele în format text simplu	Valorile monetare se trimit în unități minore. Exemplu: 1000 înseamnă 10.00, iar 255 înseamnă 2.55.
Sumele în format INP	Valorile monetare se trimit zecimal normal, de exemplu 10.00 sau 2.55.
Bon fiscal valid	Un bon fiscal trebuie să conțină cel puțin un articol de vânzare sau storno.
Selecția dispozitivului	Dacă există mai multe dispozitive fiscale, este recomandat să se transmită explicit ID-ul fiscal.

Exemple rapide de interpretare a sumelor în format text simplu:

POS^1000 înseamnă o tranzacție POS de 10.00.

I^5000 înseamnă cash in de 50.00.

DV^125 înseamnă discount valoric de 1.25.

Recomandare de implementare

Dacă sistemul integrator are deja valori zecimale, este bine să existe o funcție centrală de conversie către unități minore pentru formatul text simplu, pentru a evita erorile de rotunjire.

3. Modelul de răspuns

În funcție de canalul de integrare, răspunsul este livrat fie în fișier text, fie în JSON, fie ca extra-uri Android. Din punct de vedere funcțional, integratorul trebuie să verifice în primul rând indicatorul de succes și apoi să citească detaliile operației.

Canal	Indicator primar	Detalii uzuale
Fișier de răspuns	BONOK=1 sau BONOK=0	NRBON, NRZ, ERRCODE, ERRINFO
API HTTP	ReceiptStatus true sau false	ReceiptNumber, ReceiptInfo, ErrorCode, ErrorInfo
Android Intent	SUCCESS true sau false	RESPONSE_JSON, HTTP_STATUS_CODE, ERROR_INFO

Exemplu fișier de răspuns - succes

```
BONOK=1
NRBON=123
NRZ=456
```

Exemplu fișier de răspuns - eroare

```
BONOK=0
ERRCODE=DEVICE_NOT_FOUND
ERRINFO=Dispozitivul cerut nu există.
```

Exemplu răspuns API / Intent - succes

```
{
  "ReceiptStatus": true,
  "ReceiptNumber": "123",
  "ReceiptInfo": "BF:123;Z:456",
  "ErrorCode": "",
  "ErrorInfo": ""
}
```

Exemplu răspuns API / Intent - eroare

```
{
  "ReceiptStatus": false,
  "ReceiptNumber": "0",
  "ReceiptInfo": "",
  "ErrorCode": "DEVICE_NOT_FOUND",
  "ErrorInfo": "Dispozitivul cerut nu există."
}
```

4. Windows: integrare prin fișier de comandă

Integrarea prin fișier de comandă este metoda tradițională și rămâne utilă în proiectele unde aplicația integratoare poate scrie fișiere într-un folder local, dar nu poate sau nu dorește să facă apeluri HTTP.

Folder	Rol în flux
Bonuri	Conține comenzile noi care urmează să fie prelucrate.
Istoric	Conține fișierele mutate după procesare, pentru audit sau arhivare.
Raspuns	Conține fișierul de rezultat produs de aplicație după prelucrare.

Se creează un fișier cu extensia .txt, .inp sau .json.
Se scrie comanda în formatul dorit.
Fișierul se copiază în folderul Bonuri sau în folderul dedicat dispozitivului.
Aplicația detectează fișierul, îl execută și îl mută în Istoric.
Integratorul citește fișierul rezultat din folderul Raspuns.

Exemplu .txt - STATUS

```
STATUS
```

Exemplu .txt - POS

```
POS^1000
```

Exemplu .txt - bon fiscal simplu

```
ID^2
S^APA^500^1000^buc^1^1
P^1^500
```

Practică recomandată

Pentru depanare, păstrați întotdeauna perechea fișier comandă - fișier răspuns. Aceasta este cea mai rapidă metodă de reconstrucție a unui incident raportat de client.

5. Formatul text simplu: structură generală

Formatul text simplu folosește o convenție foarte stabilă: codul comenzii, urmat de separatorul ^ și de parametrii acelei comenzi. O comandă complexă, cum este emiterea unui bon fiscal, se compune din mai multe linii care descriu contextul, articolele, subtotalul și plata.

Categorie	Comenzi incluse	Scop
Context	ID, CF, NF, TL	Selectează dispozitivul, codul fiscal al clientului, tipul de bon și textul liber.
Articole	S, VS	Adaugă produse normale sau produse storno.
Subtotal și ajustări	ST, DP, DV, MP, MV	Aplică subtotal, discounturi și majorări.
Plată și ieșire tipărită	CB, P	Adaugă coduri de bare și linii de plată.
Operații administrative	X, Z, STATUS, BONINFO, RAPZINFO, POS, I, O, DISPLAY, MSG, D	Execută rapoarte, interogări, operații POS, cash in/out și mesaje pe display.

Exemplu complet de bon fiscal - format text simplu

```
ID^2
CF^RO123456789
TL^Mesaj de început
S^APA^500^1000^buc^1^1
DP^100
S^SUC^700^1000^buc^1^1
TL^Text asociat produsului curent
ST^
DV^50
CB^5940000000820^2
TL^Mesaj final
P^1^1150
```

6. Formatul text simplu: referință completă a comenzilor

Cod	Sintaxă	Descriere funcțională
ID	ID^{deviceId}	Selectează explicit dispozitivul fiscal pe care se execută liniile următoare.
CF	CF^{codFiscal}	Setează codul fiscal al clientului pentru bonul fiscal curent.
NF	NF sau NF^{text}	Inițializează un bon nefiscal. Textul opțional devine primul mesaj tipărit.
TL	TL^{text}	Adaugă text liber. Destinația exactă depinde de poziția comenzii în flux.
S	S^{nume}^{pret}^{cantitate}^{um}^{tva}^{departament}	Adaugă un articol de vânzare.
VS	VS^{nume}^{pret}^{cantitate}^{um}^{tva}^{departament}	Adaugă un articol storno.
ST	ST^	Deschide contextul de subtotal pentru ajustări la nivel de bon.
DP	DP^{valoare}	Aplică discount procentual pe articol sau pe subtotal, în funcție de context.
DV	DV^{valoare}	Aplică discount valoric pe articol sau pe subtotal, în funcție de context.
MP	MP^{valoare}	Aplică majorare procentuală pe articol sau pe subtotal, în funcție de context.
MV	MV^{valoare}	Aplică majorare valorică pe articol sau pe subtotal, în funcție de context.
CB	CB^{valoare}^{tip}	Adaugă un barcode sau un cod QR pentru tipărire.
P	P^{tipPlata}^{valoare?}	Adaugă o linie de plată. Valoarea este opțională în anumite scenarii.
X	X	Execută raport X.
Z	Z	Execută raport Z.
STATUS	STATUS	Interoghează starea dispozitivului fiscal.
BONINFO	BONINFO^{raport}^{bon}	Citește informațiile unui bon din jurnal.
RAPZINFO	RAPZINFO^{start}^{end}	Citește un interval de rapoarte Z din jurnal.
POS	POS^{suma}^{idPos?}	Inițiază o tranzacție POS. ID-ul POS este opțional.
I	I^{suma}	Execută cash in.
O	O^{suma}	Execută cash out.
DISPLAY / MSG / D	DISPLAY^{linie?}^{text} MSG^{linie?}^{text} D^{linie?}^{text}	Trimite un mesaj către display-ul client/operator, dacă dispozitivul suportă operația.

7. Formatul text simplu: comenzi de context

Comenzile de context sunt cele care stabilesc cadrul în care vor fi interpretate celelalte linii. O integrare stabilă pornește aproape întotdeauna cu definirea corectă a acestui context.

Comandă	Câmpuri	Detalii de utilizare
ID	deviceId	Se folosește atunci când există mai multe dispozitive fiscale și integratorul trebuie să indice explicit dispozitivul țintă.
CF	codFiscal	Se folosește pentru transmiterea codului fiscal al clientului pe bonul fiscal.
NF	text opțional	Transformă documentul într-un bon nefiscal. Dacă include text, acel text apare ca primă linie informativă.
TL	text	Plasează text înainte de produse, pe articolul curent sau după subtotal, în funcție de poziție.

Exemplu - bon nefiscal

```
NF^Document informativ
TL^Acesta nu este bon fiscal
TL^Mulțumim pentru vizită
```

Regulă importantă pentru TL

TL nu are o poziție fixă. Dacă este înaintea primului produs, se comportă ca text de început. Dacă este imediat după un produs, se asociază produsului curent. Dacă este după subtotal, devine text de final.

8. Formatul text simplu: comenzi de articol

Comenzile S și VS descriu structura comercială a bonului. Ele sunt cele mai importante linii într-un bon fiscal, deoarece definesc produsul, prețul, cantitatea și tratamentul TVA.

Poziție	Semnificație	Exemplu
1	Nume articol	APA
2	Preț unitar în unități minore	500 pentru 5.00
3	Cantitate	1000 pentru 1.000
4	Unitate de măsură	buc
5	Cota / grupa TVA	1
6	Departament	1

Diferența dintre S și VS este comportamentul fiscal: S adaugă o vânzare normală, iar VS adaugă o poziție storno. În rest, ordinea câmpurilor este aceeași.

Exemplu - articol normal și articol storno

```
S^APA^500^1000^buc^1^1
VS^RETUR APA^500^1000^buc^1^1
```

9. Formatul text simplu: subtotal, discounturi și majorări

Comenzile ST, DP, DV, MP și MV trebuie citite împreună. Ele formează mecanismul de ajustare a valorilor. Locul în care apar în comandă determină dacă ajustarea se aplică pe articol sau pe bon.

Comandă	Nivel posibil	Interpretare
ST	Bon	Deschide zona de subtotal. După această linie, ajustările devin ajustări de bon.
DP	Articol sau bon	Discount procentual.
DV	Articol sau bon	Discount valoric.
MP	Articol sau bon	Majorare procentuală.
MV	Articol sau bon	Majorare valorică.

Dacă DP, DV, MP sau MV apare imediat după un articol, ajustarea se aplică articolului curent.
Dacă aceeași comandă apare după ST, ajustarea se aplică subtotalului bonului.

Exemplu - discount pe articol și discount pe bon

```
S^APA^500^1000^buc^1^1
DP^100
S^SUC^700^1000^buc^1^1
ST^
DV^50
P^1^1050
```

10. Formatul text simplu: plăți, coduri de bare și operații administrative

Această familie de comenzi acoperă închiderea bonului, tipărirea informațiilor suplimentare și operațiile operaționale care nu reprezintă articole comerciale.

Comandă	Scop	Observații
CB	Tipărire barcode sau QR	Tipul determină standardul de codificare.
P	Plată	Tipul plății este numeric și poate include sau nu valoare explicită.
X / Z	Rapoarte fiscale	X este raport intermediar, Z este raport de închidere.
STATUS	Interogare stare	Util pentru testarea conectivității și diagnoză.
BONINFO / RAPZINFO	Citire jurnal	Permit interogarea informațiilor istorice.
POS	Plată bancară	Poate fi lansată din orice canal de integrare.
I / O	Cash in / cash out	Lucrează cu sume în unități minore.
DISPLAY / MSG / D	Mesaje pe display	Disponibilitatea depinde de modelul dispozitivului.

Tip plată	Semnificație
1	Cash
2	Card
3	Credit
4	FoodStamp
5	ValueTichets
6	Voucher
7	Modern
8	Other1
9	Other2

Tip barcode	Semnificație
1	EAN8
2	EAN13
3	CODE128
4	QR

Exemple - operații administrative

```
STATUS
X
Z
BONINFO^123^45
RAPZINFO^120^125
POS^1000
POS^1000^7
I^5000
O^1000
DISPLAY^1^Bun venit
```

11. Formatul INP: reguli generale

Formatul INP este orientat către compatibilitate cu aplicații care generează deja structuri tip linie logică. Spre deosebire de formatul text simplu, valorile monetare și cantitățile se trimit zecimal explicit.

Regulă	Explicație
Același număr logic pe tot documentul	Toate liniile unei comenzi INP trebuie să folosească același număr logic.
Număr logic valid	Numărul logic trebuie să fie între 1 și 99.
Ordine strictă a liniilor	La fel ca în formatul text simplu, ordinea liniilor este relevantă pentru context.
Plata închide fluxul comercial	După o linie de tip T nu se mai recomandă alte operații comerciale, cu excepția secvențelor compatibile de final.
Sume și cantități zecimale	Exemplu: 12.50 și 2.000 sunt valori valide în INP.

Exemplu complet - bon fiscal în INP

```
K,2,______,_,__;RO123456789;
P,2,______,_,__;Mesaj început;;;;;
S,2,______,_,__;Produs test;12.50;2.000;1;1;19%;0;0;buc;
W,2,______,_,__;bc;Comentariu produs;
L,2,______,_,__;;;;;;
Q,2,______,_,__;2;5940000000820;
T,2,______,_,__;0;25.00;;;;
```

12. Formatul INP: referință completă a comenzilor

Cod	Rol	Observație tehnică
K	Cod fiscal client	Inițializează codul fiscal al clientului pentru bonul fiscal.
S	Articol de vânzare	Definește produsul și valorile comerciale principale.
C	Discount sau majorare	Tipul intern stabilește dacă este procent sau valoare și dacă este reducere sau creștere.
L	Subtotal	Deschide zona de subtotal.
P	Text liber	Poate funcționa ca text general sau contextual.
W	Text suplimentar	Se utilizează frecvent ca text asociat unui articol.
Q	Barcode / QR	Tipul determină standardul de codificare.
T	Plată	Închide partea comercială a fluxului printr-o formă de plată.
Z	Raport X sau Z	Parametrul transmis diferențiază între cele două variante.
I	Cash in / cash out	Tipul de acțiune determină sensul operației.

13. Formatul INP: câmpuri și coduri speciale

Comandă	Cod / câmp	Semnificație
C	0	Majorare procentuală
C	1	Discount procentual
C	2	Majorare valorică
C	3	Discount valoric
Q	1	EAN8
Q	2	EAN13
Q	3	CODE128
Q	4	QR
T	0	Cash; mapat intern la tipul fiscal 1
T	1	Card; mapat intern la tipul fiscal 2
T	2	Credit
T	3	FoodStamp
T	4	ValueTichets
T	5	Voucher
T	6	Modern
T	7	Other1
T	8	Other2
I	0	Cash in
I	1	Cash out
Z	0	Raport X
Z	altă valoare / lipsă	Raport Z

În formatul INP, comenzile P și W sunt dependente de context, la fel ca TL în formatul text simplu. Dacă urmează imediat după un articol, ele devin text asociat articolului. Dacă apar în afara acelei zone, se comportă ca text general de bon.

Bon nefiscal în INP

Dacă documentul conține numai text și coduri de bare, fără articole comerciale, aplicația poate interpreta fluxul ca document nefiscal.

14. API HTTP: endpoint-uri și contract de apel

API-ul HTTP este recomandat pentru integrări moderne. El permite atât apeluri simple sincron, cât și execuție asincronă cu job-uri. În majoritatea proiectelor noi, acesta este canalul preferat pe Windows.

Endpoint	Metodă	Descriere
/api/health	GET	Verifică dacă serviciul răspunde.
/api/devices	GET	Listează dispozitivele disponibile.
/api/devices/{deviceId}	GET	Returnează detaliile unui dispozitiv.
/api/receipt	POST	Execută o comandă generală.
/api/devices/{deviceId}/receipt	POST	Execută o comandă pe un dispozitiv fiscal anume.
/api/jobs	GET	Listează job-urile async.
/api/jobs/{jobId}	GET	Returnează starea și rezultatul unui job async.

Corpul cererii poate fi trimis în două forme uzuale: text simplu multi-linie sau listă JSON de linii. În ambele cazuri, conținutul logic este același.

Exemplu body - text simplu

```
STATUS
```

Exemplu body - JSON array

```
[
  "ID^2",
  "S^APA^500^1000^buc^1^1",
  "P^1^500"
]
```

15. API HTTP: exemple de integrare

PowerShell - STATUS

```
$body = "STATUS"
Invoke-RestMethod `
  -Method Post `
  -Uri "http://localhost:65400/api/receipt" `
  -ContentType "text/plain" `
  -Body $body
```

cURL - POS

```
curl -X POST "http://localhost:65400/api/receipt" ^
  -H "Content-Type: text/plain" ^
  --data-binary "POS^1000"
```

cURL - bon fiscal pe dispozitiv explicit

```
curl -X POST "http://localhost:65400/api/devices/2/receipt" ^
  -H "Content-Type: text/plain" ^
  --data-binary "S^APA^500^1000^buc^1^1
P^1^500"
```

JavaScript / Node.js

```
const payload = "STATUS";

const response = await fetch("http://localhost:65400/api/receipt", {
  method: "POST",
  headers: { "Content-Type": "text/plain" },
  body: payload
});

const data = await response.json();
console.log(data);
```

C# / HttpClient

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

var payload = "POS^1000";
using var client = new HttpClient();
var content = new StringContent(payload, Encoding.UTF8, "text/plain");
var response = await client.PostAsync("http://localhost:65400/api/receipt", content);
var body = await response.Content.ReadAsStringAsync();
Console.WriteLine(body);
```

16. API HTTP: execuție asincronă

Modul asincron este util când integratorul nu dorește să țină conexiunea deschisă până la terminarea operației sau când vrea să trateze execuția ca job de fundal.

Comanda se trimite cu parametrul de query ?async=1 sau cu header-ul X-FiscalNet-Async: true.

Aplicația răspunde cu acceptarea job-ului și identificatorul acestuia.

Integratorul citește ulterior rezultatul din /api/jobs/{jobId}.

Exemplu apel async

```
curl -X POST "http://localhost:65400/api/receipt?async=1" ^
  -H "Content-Type: text/plain" ^
  --data-binary "POS^1000"
```

Exemplu citire job

```
curl "http://localhost:65400/api/jobs/JOB_ID_AICI"
```

17. Android: contractul Intent

Pe Android, integrarea se face prin Intent explicit către pachetul FiscalNet. Sunt două acțiuni standard: una pentru inițializare și una pentru procesarea propriu-zisă a unei comenzi.

Element	Valoare
Package	com.econmedia.fiscalnet
Action pentru warm-up	com.econmedia.fiscalnet.action.WARMUP_APP
Action pentru procesare comandă	com.econmedia.fiscalnet.action.PROCESS_COMMAND
Extra comandă text	com.econmedia.fiscalnet.extra.COMMAND_TEXT
Extra comandă linii	com.econmedia.fiscalnet.extra.COMMAND_LINES
Extra sursă	com.econmedia.fiscalnet.extra.SOURCE
Extra referință	com.econmedia.fiscalnet.extra.REFERENCE
Extra timeout	com.econmedia.fiscalnet.extra.TIMEOUT_MS
Extra mutare în fundal	com.econmedia.fiscalnet.extra.MOVE_TO_BACK

Pentru a primi un răspuns înapoi, aplicatia apelantă trebuie să folosească ActivityResultLauncher sau un mecanism

echivalent de tip startActivityForResult. Un simplu startActivity nu este suficient dacă integrarea are nevoie de rezultat.

18. Android: exemple de apel și interpretarea răspunsului

Kotlin - PROCESS_COMMAND

```
private val fiscalLauncher = registerForActivityResult(
    ActivityResultContracts.StartActivityForResult()
) { result ->
    val data = result.data
    val responseJson = data?.getStringExtra("com.econmedia.fiscalnet.result.RESPONSE_JSON")
    val success = data?.getBooleanExtra("com.econmedia.fiscalnet.result.SUCCESS", false) ?: false
    val httpStatus = data?.getIntExtra("com.econmedia.fiscalnet.result.HTTP_STATUS_CODE", 0) ?: 0
}

val intent = Intent("com.econmedia.fiscalnet.action.PROCESS_COMMAND").apply {
    `package` = "com.econmedia.fiscalnet"
    putExtra("com.econmedia.fiscalnet.extra.COMMAND_TEXT", "POS^1000")
    putExtra("com.econmedia.fiscalnet.extra.SOURCE", "MyAndroidApp")
    putExtra("com.econmedia.fiscalnet.extra.REFERENCE", "ORDER-10025")
    putExtra("com.econmedia.fiscalnet.extra.TIMEOUT_MS", 20000)
}

fiscalLauncher.launch(intent)
```

Kotlin - WARMUP_APP

```
private val warmupLauncher = registerForActivityResult(
    ActivityResultContracts.StartActivityForResult()
) { result ->
    val data = result.data
    val initialized = data?.getBooleanExtra("com.econmedia.fiscalnet.result.INITIALIZED", false) ?:
false
    val infrastructureReady =
data?.getBooleanExtra("com.econmedia.fiscalnet.result.INFRASTRUCTURE_READY", false) ?: false
    val backgrounded = data?.getBooleanExtra("com.econmedia.fiscalnet.result.BACKGROUNDED", false) ?:
false
}

val intent = Intent("com.econmedia.fiscalnet.action.WARMUP_APP").apply {
    `package` = "com.econmedia.fiscalnet"
    putExtra("com.econmedia.fiscalnet.extra.MOVE_TO_BACK", true)
    putExtra("com.econmedia.fiscalnet.extra.TIMEOUT_MS", 20000)
}

warmupLauncher.launch(intent)
```

ADB - test Intent Android

```
adb shell am start ^
  -a com.econmedia.fiscalnet.action.PROCESS_COMMAND ^
  -p com.econmedia.fiscalnet ^
  --es com.econmedia.fiscalnet.extra.COMMAND_TEXT "STATUS" ^
  --es com.econmedia.fiscalnet.extra.SOURCE "ADB test"
```

Extra de răspuns	Utilitate
com.econmedia.fiscalnet.result.SUCCESS	Indicator principal de succes.
com.econmedia.fiscalnet.result.RESPONSE_JSON	Răspunsul complet, în format logic identic cu API-ul.
com.econmedia.fiscalnet.result.HTTP_STATUS_CODE	Cod orientativ pentru tratarea tehnică a rezultatului.
com.econmedia.fiscalnet.result.RECEIPT_STATUS	Indicatorul funcțional de emitere bon.
com.econmedia.fiscalnet.result.RECEIPT_NUMBER	Numărul bonului, dacă există.
com.econmedia.fiscalnet.result.RECEIPT_INFO	Informații sumare despre rezultat.
com.econmedia.fiscalnet.result.ERROR_CODE	Codul erorii, dacă operația a eșuat.
com.econmedia.fiscalnet.result.ERROR_INFO	Mesajul de eroare pentru integrator.

19. Reguli speciale pentru comanda POS

Comanda POS are un comportament particular, deoarece poate implica lansarea unei integrări bancare sau simularea pozitivă a răspunsului în anumite scenarii controlate.

Comanda POS poate fi transmisă prin fișier, API HTTP sau Android Intent.

POS se poate executa din orice punct de integrare, inclusiv atunci când nu este trimis explicit un ID fiscal în corpul comenzii.

Dacă dispozitivul fiscal din context este un dispozitiv de test, nu este obligatoriu să existe un POS bancar programat.

Dacă dispozitivul fiscal din context nu are licență Plus, aplicația poate întoarce direct un răspuns pozitiv fără a trimite operația către un POS real.

Dacă există mai multe dispozitive bancare POS, este recomandat să se trimită explicit și ID-ul POS-ului bancar.

Exemple POS

```
POS^1000
POS^1000^7
```

Recomandare pentru producție

Pentru un comportament predictibil în medii cu mai multe dispozitive, transmiteți explicit atât ID-ul dispozitivului fiscal, cât și ID-ul POS-ului bancar atunci când este disponibil.

20. Probleme frecvente și listă de verificare

Situație	Ce trebuie verificat
Nu apare fișierul de răspuns	Folderul corect, drepturile de scriere, existența aplicației în execuție și formatul fișierului trimis.
ReceiptStatus este false	Citiți ErrorCode și ErrorInfo înainte de orice alt pas de depanare.
API-ul răspunde cu eroare tehnică	Adresa endpoint-ului, metoda HTTP, Content-Type și corpul cererii.
Intent Android nu întoarce rezultat	Verificați că s-a folosit un launcher de tip Activity Result și nu doar un start simplu.
POS nu ajunge la dispozitivul bancar	Verificați dacă dispozitivul fiscal este de test, dacă licența este Plus și dacă există un POS bancar disponibil.
Dispozitivul fiscal greșit este folosit	Transmiteți explicit ID-ul dispozitivului fiscal.
Bonul este respins la parsare	Verificați ordinea liniilor și prezența a cel puțin unui articol într-un bon fiscal.

21. Anexă: exemple scurte de referință

Fișier text - STATUS

```
STATUS
```

Fișier text - bon fiscal minim

```
S^APA^500^1000^buc^1^1
P^1^500
```

INP - bon fiscal minim

```
S,2,______,_,__;Produs;5.00;1.000;1;1;19%;0;0;buc;
T,2,______,_,__;0;5.00;;;;
```

API - STATUS

```
curl -X POST "http://localhost:65400/api/receipt" -H "Content-Type: text/plain" --data-binary "STATUS"
```

Android Intent - PROCESS_COMMAND

```
Intent("com.econmedia.fiscalnet.action.PROCESS_COMMAND")
  .setPackage("com.econmedia.fiscalnet")
  .putExtra("com.econmedia.fiscalnet.extra.COMMAND_TEXT", "STATUS")
```

Fișier text - POS

```
POS^1000
```

API - POS

```
curl -X POST "http://localhost:65400/api/receipt" -H "Content-Type: text/plain" --data-binary "POS^1000"
```

Android Intent - WARMUP_APP

```
Intent("com.econmedia.fiscalnet.action.WARMUP_APP")
  .setPackage("com.econmedia.fiscalnet")
  .putExtra("com.econmedia.fiscalnet.extra.MOVE_TO_BACK", true)
  .putExtra("com.econmedia.fiscalnet.extra.TIMEOUT_MS", 20000)
```

Exemplu răspuns JSON - succes

```
{
  "ReceiptStatus": true,
  "ReceiptNumber": "123",
  "ReceiptInfo": "BF:123;Z:456",
  "ErrorCode": "",
  "ErrorInfo": ""
}
```