



GetResults() — Return Format

Coding Results & Grouper Integration

Supported Tariff Systems

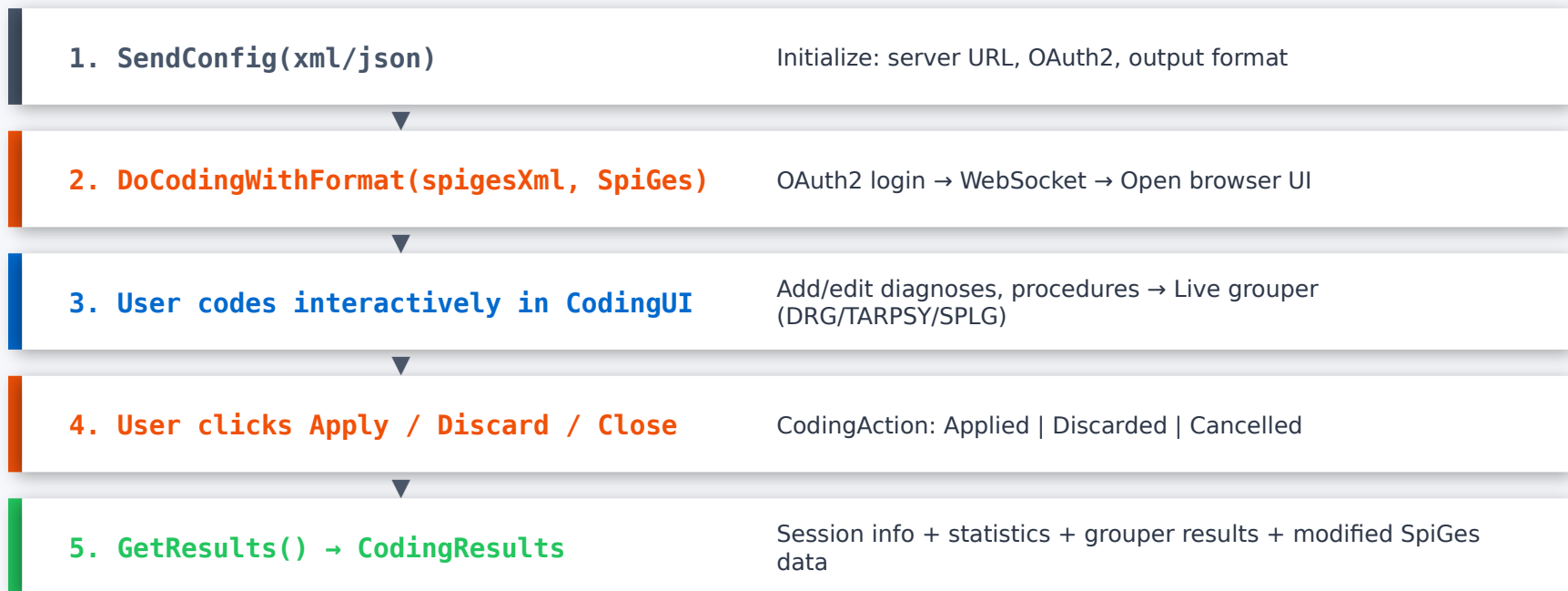
Current production + roadmap

Tariff	Status	Grouper	Since
SwissDRG	☐ Production	SwissDRG Grouper	2026
TARPSY	☐ Production	TARPSY Grouper	2026
ST Reha	☐ Production	ST Reha Grouper	2026
SPLG	☐ Planned	SPLG Grouper	Q2 2026

SPLG (ambulante Pauschalen) wird als neuer Tarif in codingResults[] integriert.
Die bestehende Struktur bleibt abwärtskompatibel.

Integration Flow

HIS → KodeMed DLL → CodingUI → Results



Two Integration Modes

Interactive (UI) vs Headless (batch)

Interactive Mode

`DoCodingWithFormat()`

- OAuth2 authentication
- WebSocket session to server
- Embedded browser (WebView2)
- User codes in CodingUI
- Live grouper results
- Apply / Discard / Cancel

⚠ BLOCKING: holds calling thread

Use: Daily coding workflow

Headless Mode

`DoCoding()`

- No browser / no UI
- Data validation
- Server-side grouping
- Batch processing
- Returns CodingResults

Use: Batch import, nightly jobs

Both modes return the same CodingResults structure via GetResults()

Why CodingClient for REST/WebSocket?

DoCodingWithFormat() is blocking — CodingClient decouples the HIS

Option A: COM DLL (blocking)

HIS thread

- DoCodingWithFormat() —
- **BLOCKED 2-15 min** —
- returns result —

⚠ **HIS UI freezes during session**

Options B & C: CodingClient

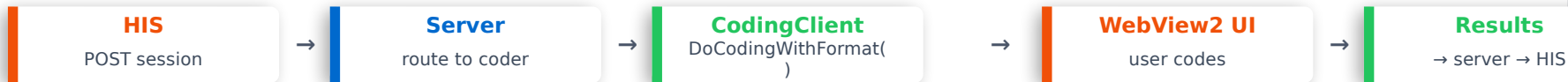
HIS thread

- POST /coding/session —
- **returns immediately** —

CodingClient (coder's PC)

- DoCodingWithFormat() — blocked here, not in HIS

REST/WebSocket Integration Flow via CodingClient



□ HIS stays responsive • Blocking runs on coder's workstation • Results via poll, webhook, or WS event

GetResults() — Response Structure

Root object: CodingResults

Field	Type	Description
sessionId	string	Unique coding session ID
success	bool	Processing succeeded
codingAction	enum	Applied(1) Discarded(2) Cancelled(3) Timeout(4) Error(5)
resultData	string	Final SpiGes XML/JSON (modified if Applied, original otherwise)
originalData	string	Original data before coding
modifiedData	string	Modified data (only set if Applied)
casesProcessed	int	Number of cases processed
diagnosesCount	int	Total diagnoses across all cases
treatmentsCount	int	Total treatments across all cases
warnings	string[]	Processing warnings

codingResults[] – Grouper Results Per Case

One entry per case with DRG/tariff data

Field	Type	Description
tariff	string	SWISSDRG TARPSY STREHA SPLG (future)
drgCode	string	Assigned DRG/PCG/RCG code (e.g. I68A)
costWeight	decimal	Cost weight
effectiveCostWeight	decimal	Effective CW after LOS adjustments
pccl	int	Patient Clinical Complexity Level (0-4)
los / ltp / htp / alos	int/decimal	Length of stay, low/high trim, average LOS
mainDiagnosis	string	Main diagnosis (ICD code)
procedures	string[]	CHOP procedure codes
supplements[]	object[]	Additional payments (code, amount, count)
grouperVersion	string	Grouper version used
groupingSuccess	bool	Whether grouping succeeded

SPLG — Planned Extension

Spitalplanungsleistungsgrouper integration (Q2 2026)

Current: Inpatient

- SwissDRG — Acute care
- TARPSY — Psychiatry
- ST Reha — Rehabilitation

```
tariff: "SWISSDRG" | "TARPSY" | "STREHA"
```

Future: + SPLG

- SPLG — Spitalplanungsleistungsgrouper

New fields:

```
splgCode, splgVersion  
ambulatory, baseRate  
minutesOR, complexity
```

```
tariff: "SPLG"
```

Backwards compatible: existing integrations continue to work unchanged. SPLG adds new fields to codingResults[.]

Example Response (JSON)

GetResults() after successful coding session

```
{
  "sessionId": "a1b2c3d4-e5f6-7890",
  "success": true,
  "codingAction": 1, // Applied
  "applied": true,
  "casesProcessed": 1,
  "resultData": "<Unternehmen>...</Unternehmen>",
  "codingResults": [{
    "fallId": 12345,
    "tarif": "SWISSDRG",
    "drgCode": "I68A",
    "costWeight": 1.245,
    "effectiveCostWeight": 1.312,
    "pcc1": 2,
    "los": 7, "ltp": 2, "htp": 18,
    "mainDiagnosis": "I63.9",
    "procedures": ["39.74.11", "99.BA.11"],
    "grouperVersion": "13.0",
    "groupingSuccess": true,
    "supplements": [{ "code": "ZE-2026-01.01", "amount": 1250.00 }]
  }]
}
```

Example Response (XML)

GetResults() with outputFormat=xml

```
<CodingResults>
  <SessionId>a1b2c3d4-e5f6-7890</SessionId>
  <Success>>true</Success>
  <CodingAction>Applied</CodingAction>
  <Applied>>true</Applied>
  <CasesProcessed>1</CasesProcessed>
  <DiagnosesCount>4</DiagnosesCount>
  <TreatmentsCount>2</TreatmentsCount>
  <ResultData>&lt;Unternehmen&gt;...&lt;/Unternehmen&gt;</ResultData>
  <CodingResults>
    <Result fall_id="12345">
      <Tariff>SWISSDRG</Tariff>
      <DrgCode>I68A</DrgCode>
      <CostWeight>1.245</CostWeight>
      <EffectiveCostWeight>1.312</EffectiveCostWeight>
      <Pccl>2</Pccl>
      <Los>7</Los><Ltp>2</Ltp><Htp>18</Htp>
      <MainDiagnosis>I63.9</MainDiagnosis>
      <GrouperVersion>13.0</GrouperVersion>
      <GroupingSuccess>true</GroupingSuccess>
    </Result>
  </CodingResults>
</CodingResults>
```

Roadmap

GetResults() evolution

Timeline	Feature
Current	SwissDRG, TARPSY, ST Reha grouper results
Q2 2026	SPLG Spitalplanungsleistungsgrouper (integration test ready)
Q4 2026	Multi-tariff per case (DRG + SPLG)
2027	Real-time grouper suggestions during coding

KodeMed

Questions?

info@kodemed.ch

<https://kodemed.ch>