



API REFERENCE · API V1

Balance

Check how much you can send before you send it, and alert your team well before the balance runs out.

Updated 2026-10-10 · generated 2026-10-10 · smsray.com/docs/balance

1 Retrieve balance

GET /sms/balance <https://api.smsray.com/api/sms/v1/sms/balance>

Read your workspace balance, per-type rates, allowed message types and this API client's counters.

HEADERS

x-api-key string **required**

Your secret key, ls_live_ followed by 48 hex characters.

EXAMPLE REQUESTS

cURL

```
curl https://api.smsray.com/api/sms/v1/sms/balance \
-H "x-api-key: $SMSRAY_API_KEY"
```

Node.js

```
const res = await fetch("https://api.smsray.com/api/sms/v1/sms/balance", {
  headers: { "x-api-key": process.env.SMSRAY_API_KEY },
});
const { balance, rate, workspaceStatus } = await res.json();
if (balance < rate.otp * 100) {
  // alert your team before OTPs start failing with 402
}
```

Python

```
import os, requests

r = requests.get(
    "https://api.smsray.com/api/sms/v1/sms/balance",
    headers={"x-api-key": os.environ["SMSRAY_API_KEY"]},
    timeout=10,
).json()
print(r["balance"], r["workspaceStatus"])
```

PHP

```
$ch = curl_init("https://api.smsray.com/api/sms/v1/sms/balance");
curl_setopt_array($ch, [
    CURLOPT_RETURNTRANSFER => true,
    CURLOPT_HTTPHEADER => ["x-api-key: " . getenv("SMSRAY_API_KEY")],
]);
$res = json_decode(curl_exec($ch), true);
echo $res["balance"] . " " . $res["workspaceStatus"];
```

Go

```
// imports: encoding/json, fmt, log, net/http, os
req, _ := http.NewRequest("GET", "https://api.smsray.com/api/sms/v1/sms/balance", nil)
req.Header.Set("x-api-key", os.Getenv("SMSRAY_API_KEY"))

res, err := http.DefaultClient.Do(req)
if err != nil {
    log.Fatal(err)
}
defer res.Body.Close()

var out struct {
    Balance          float64 `json:"balance"`
    WorkspaceStatus string  `json:"workspaceStatus"`
}
json.NewDecoder(res.Body).Decode(&out)
fmt.Println(out.Balance, out.WorkspaceStatus)
```

RESPONSES

200

```
{
  "balance": 5000.0,
  "rate": { "transactional": 0.5, "otp": 0.5, "promotional": 0.5 },
  "allowedTypes": ["transactional", "otp"],
  "counters": { "sent": 120, "delivered": 110, "failed": 4 },
  "workspaceStatus": "active"
}
```

RESPONSE FIELDS

balance number **required**

Credit remaining for the current monthly period, in your workspace's currency (NPR for Nepal workspaces), shared by every API client in the workspace.

rate object **required**

Price per SMS segment for transactional, otp and promotional. Cost = rate × segments. promotional is enabled only on Enterprise; check allowedTypes.

allowedTypes string[] **required**

Message types this workspace may send. Others return 403 forbidden.

counters object **required**

This API client's own sent, delivered and failed totals.

workspaceStatus string **required**
active, pending or suspended.

ERRORS

HTTP	CODE	WHEN
401	unauthorized	The x-api-key header is missing ("Missing x-api-key") or the key is unknown ("Invalid API key").
403	client_disabled	The API client that owns this key has been disabled in the dashboard.
429	rate_limited	You exceeded your per-client request rate (default 20 requests per second). Honour Retry-After.
500	server_error	Something went wrong on our side. The body never contains a stack trace.

NOTES

- GET requests keep working while your workspace is **pending**, so you can integrate and check your setup before activation.
- Your rate is your plan's per-SMS price. See [pricing](#). Remaining balance expires at the end of each monthly period; top-ups are charged at your plan rate.
- When the balance is lower than a message's cost, the send returns 402 **insufficient_balance**.
- A message's cost is refunded only when delivery finally fails (**failed**). **undelivered** is not refunded.

The web page is always the current version: smsray.com/docs/balance

Full API reference PDF: smsray.com/docs/pdf · Your keys and live examples: app.smsray.com/developers