



GUIDES · API V1

Numbers & countries

Send numbers in E.164 and let the API tell you where you can deliver.

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

Number formats

E.164 is recommended: a +, the country code and the subscriber number, with no spaces, for example +9779801234567. The national format of your workspace's country is accepted too, and the country code without + works as well.

YOU SEND	READ AS	NOTES
+9779801234567	+9779801234567	E.164 — recommended.
9779801234567	+9779801234567	Country code without +.
9801234567	+9779801234567	National format of the workspace's country.
014XXXXXX (landline)	—	400 <code>invalid_number</code> : can't receive SMS.

Every number must match `^\+?\d{7,15}$`. SMSRay stores recipients in E.164, so to in message lookups and webhooks is always `+<country code><number>`.

Where you can send

Your workspace has a country, set when it is created. You can send to that country once SMSRay delivers there. Today **Nepal is live**; **India** and **Bangladesh** are coming soon. Ask the API for the current list with [GET /sms/coverage](#) instead of hard-coding it.

Number errors

HTTP	CODE	WHEN
400	<code>invalid_request</code>	to is missing, shorter than 7 digits or not digits (zod details).
400	<code>invalid_number</code>	The number can't receive SMS — for example a landline, or a number that fails its country's mobile rules.
422	<code>destination_not_supported</code>	The number is valid, but in a country your workspace can't send to yet. details is { <code>country</code> , <code>allowed</code> }.

Show 422 differently from 400. The number isn't wrong — we just don't deliver there yet:

422 application/json

```
{
  "error": "We don't deliver to India yet. This workspace can send to: Nepal.",
  "code": "destination_not_supported",
  "details": { "country": "IN", "allowed": ["NP"] }
}
```

Node.js

```
const res = await fetch("https://api.smsray.com/api/sms/v1/sms/send", { method: "POST", headers,
»   body });
if (res.status === 422) {
  const { code, details } = await res.json();
  if (code === "destination_not_supported") {
    const name = new Intl.DisplayNames(["en"], { type: "region" }).of(details.country);
    return showError(`We don't deliver to ${name} yet.`); // details.allowed lists where you can
»   send
  }
}
```

Python

```
r = requests.post(f"{B}/sms/send", headers=H, json=payload, timeout=10)
if r.status_code == 422 and r.json()["code"] == "destination_not_supported":
    d = r.json()["details"]
    show_error(f"We don't deliver to {d['country']} yet. You can send to: {'',
»   '.join(d['allowed'])}")
```

PHP

```
$status = curl_getinfo($ch, CURLINFO_HTTP_CODE);
if ($status === 422 && $res["code"] === "destination_not_supported") {
    $country = Locale::getDisplayRegion("und_" . $res["details"]["country"], "en");
    show_error("We don't deliver to $country yet.");
}
```

Go

```
if res.StatusCode == http.StatusUnprocessableEntity {
    var e struct {
        Code    string `json:"code"`
        Details struct {
            Country string `json:"country"`
            Allowed []string `json:"allowed"`
        } `json:"details"`
    }
    json.NewDecoder(res.Body).Decode(&e)
    if e.Code == "destination_not_supported" {
        return fmt.Errorf("we don't deliver to %s yet (allowed: %v)", e.Details.Country,
»   e.Details.Allowed)
    }
}
```

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

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

SMSRAY
DEVELOPER DOCS