

# Acme Widget API — Code Examples & Formulas

## Inline Code

Every widget has a unique `widget_id` string. Pass it to the GET `/widgets/{widget_id}` endpoint to fetch details. The response content type is `application/json`.

## Python Example

Fetch a widget using the Python SDK:

```
import acme

client = acme.Client(api_key="sk-xxxx")
widget = client.widgets.get("wgt_abc123")
print(f"Widget: {widget.name}, Status: {widget.status}")
```

## JavaScript Example

The same operation in JavaScript:

```
import { AcmeClient } from "@acme/sdk";

const client = new AcmeClient({ apiKey: "sk-xxxx" });
const widget = await client.widgets.get("wgt_abc123");
console.log(`Widget: ${widget.name}, Status: ${widget.status}`);
```

## Shell Example

Or use curl directly:

```
curl -H "Authorization: Bearer sk-xxxx" \
  https://api.example.com/v1/widgets/wgt_abc123
```

## JSON Response

```
{
  "id": "wgt_abc123",
  "name": "Weather Widget",
  "status": "active",
  "created_at": "2025-06-15T10:30:00Z"
}
```

## Rate-Limit Formula

The API enforces rate limiting. Your effective request budget is  $R = \min(R_{\text{plan}}, R_{\text{burst}})$  requests per minute, where  $R_{\text{plan}}$  is your plan's base limit and  $R_{\text{burst}}$  is the burst allowance.

## Latency Model

Expected response latency follows this model:

$$L(n) = L_0 + \frac{n}{T} + \varepsilon$$

where:

- $L_0$  is the base latency (typically 20 ms)
- $n$  is the payload size in kilobytes
- $T$  is the throughput constant
- $\varepsilon$  represents network jitter

## Availability SLA

Acme guarantees an uptime of at least 99.95%. The maximum allowed monthly downtime is:

$$D_{\max} = 43200 \times (1 - 0.9995) = 21.6 \text{ seconds}$$

## Cost Calculation

The monthly cost for API usage is:

$$C = \sum_{i=1}^N p_i \times q_i$$

where  $p_i$  is the per-request price for tier  $i$  and  $q_i$  is the number of requests in that tier.