

通过本案例，大家可以了解到与 HTTP API 相关的信息。HTTP API 一方面允许用户访问 PromQL，另一方面也会开放给诸如 Grafana 等仪表盘工具或定制化报表脚本使用。除了支持 PromQL 的功能以外，HTTP API 还可以支持查找、删除时间序列的元数据管理功能。

query 函数非常适合绘图，但是不会暴露原始样本。query_range 函数可以通过在每个步骤中评估 PromQL 来工作，因此，如果在步骤之间存在多个样本，可能会丢失样本，导致输出的是具有评估步骤的时间戳而不是样本。本节提到的 HTTP API 方式还可以用于从 Prometheus 中获取原始样本，访问 `http://api/v1/query?query=up[1m]` 将返回 up 时间序列的最后 1 分钟的原始数据，用户可以手动查看这部分数据，或用脚本和代码对其进行处理（由于这是原始数据，因此无法使用 PromQL 对样本进行任何处理）。

接下来，将从 API 响应格式、表达式查询（query 和 query_range）、元数据管理、其他拓展等方面进行介绍。

5.2.1 API 响应格式

HTTP API 使用 JSON 响应格式，正如本节提供的案例一样。API 调用成功后将会返回 2xx 的 HTTP 状态码，否则可能会返回如下几种 HTTP 状态码。

- ❑ 404 Bad Request: 当参数错误或者缺失时。
- ❑ 422 Unprocessable Entity: 当表达式无法执行时。
- ❑ 503 Service Unavailable: 当请求超时或者被中断时。

JSON 响应格式如下所示。

```
{
  "status": "success" | "error",
  "data": <data>,

  // 错误信息
  "errorType": "<string>",
  "error": "<string>",
  // 如果存在不会阻止请求执行的错误，可能会返回一系列告警
  "warnings": ["<string>"]
}
```

当 API 调用成功后，Prometheus 会返回 JSON 格式的响应内容，并且在 data 节点中返回查询结果。data 节点格式如下。

```
{
  "resultType": "matrix" | "vector" | "scalar" | "string",
  "result": <value>
}
```

resultType 表示当前返回的数据类型，分为区间向量、瞬时向量、标量、字符串 4 种。返回值中 <sample_value> 占位符是样本值。因为 JSON 不支持特殊浮点值，例如 NaN、Inf