

## Общие сведения для запросов API

Текущая версия API - v1 и доступна по /api/v1/.

Формат ответа – json.

Коды ответа:

2xx – успешный запрос

400 – когда отсутствуют необходимые параметры или они не верные

422 – запрос не может быть выполнен

503 – время ожидания ответа превышено.

Метки времени могут быть переданы либо в формате [RFC3339](#), либо в формате Unix timestamp.

Метод запроса может быть как GET, так и POST. Если используется GET, то параметры запроса добавляются к url, а если POST, то параметры запроса передаются как body.

## Запросы мгновенного вектора

**Endpoint:** /api/v1/query

**Method:** GET | POST

**Параметры запроса:**

- query – строка запроса PromQL.
- time – временная метка, за которую надо выбрать данные. Если параметр не задан, берется текущее время сервера.
- timeout – максимальное время выполнения запроса. Параметр не является обязательным. Если параметр не задан, используется значение query.timeout.

Во всех примерах для выполнения запросов будет использоваться утилита curl.

Пример запроса с использованием метода GET. Запрос up, время: 19.11.2019 09:10:51. Перед выполнением запроса измените время на текущее. Время указывается в UTC:

```
curl 'http://localhost:9090/api/v1/query?query=up&time=2019-11-19T09:10:51.781Z'
```

Ответ должен быть примерно таким:

```
{
  "status":"success",
  "data":{
    "resultType":"vector",
    "result":[
      {
        "metric":{
          "__name__":"up",
          "instance":"192.168.0.12:9100",
          "job":"openstack"
        },
        "value":[
          1574154651.781,
          "1"
        ]
      },
      {
        "metric":{
          "__name__":"up",
          "instance":"192.168.0.4:9100",
          "job":"openstack"
        },
        "value":[
          1574154651.781,
          "1"
        ]
      },
      {
        "metric":{
          "__name__":"up",
          "instance":"192.168.0.7:9100",
          "job":"openstack"
        },
        "value":[
          1574154651.781,
          "1"
        ]
      },
      {
        "metric":{
```

```
    "__name__": "up",
    "instance": "localhost:9090",
    "job": "prometheus"
  },
  "value": [
    1574154651.781,
    "1"
  ]
}
]
```

Пример аналогичного запроса методом POST:

```
curl -XPOST -H 'Content-Type: application/x-www-form-urlencoded' -d "query=up" -d "time=2019-11-19T09:10:51.781Z" http://localhost:9090/api/v1/query
```

**NB!** Content-Type используется: application/x-www-form-urlencoded.

Результат ответа должен быть аналогичным предыдущему запросу.

## Запрос вектора за период времени

**Endpoint:** /api/v1/query\_range

**Method:** GET | POST

**Параметры запроса:**

- query – строка запроса PromQL.
- start – временная метка начала вектора.
- end – временная метка окончания вектора.
- step – шаг выборки данных; допустимый формат [0-9]+[smhdwy], например 5s, либо число float(секунды).
- timeout – максимальное время выполнения запроса. Необязательный параметр. Если параметр не задан, используется значение query.timeout.

**Пример.** Запрос: up, за период с 19.11.2019 10:00.50 по 19.11.2019 10:15.50, с шагом в 30 секунд:

```
curl
'http://localhost:9090/api/v1/query_range?query=up&start=2019-11-19T09:10:51.781Z&end=2019-11-19T09:15:51.781Z&step=30s'
```

Ответ должен быть примерно таким:

```
{
  "status": "success",
  "data": {
    "resultType": "matrix",
    "result": [
      {
        "metric": {
          "__name__": "up",
          "instance": "192.168.0.12:9100",
          "job": "openstack"
        },
        "values": [
          [
            1574154651.781,
            "1"
          ],
          [
            1574154681.781,
            "1"
          ],
          [
            1574154711.781,
            "1"
          ],
          [
            1574154741.781,
            "1"
          ],
          [
            1574154771.781,
            "1"
          ],
          [
            1574154801.781,
            "1"
          ],
          [
            1574154831.781,
            "1"
          ]
        ]
      }
    ]
  }
}
```

```
[
  1574154861.781,
  "1"
],
[
  1574154891.781,
  "1"
],
[
  1574154921.781,
  "1"
],
[
  1574154951.781,
  "1"
]
]
},
{
  "metric":{
    "__name__":"up",
    "instance":"192.168.0.4:9100",
    "job":"openstack"
  },
  "values":[
    [
      1574154651.781,
      "1"
    ],
    [
      1574154681.781,
      "1"
    ],
    [
      1574154711.781,
      "1"
    ],
    [
      1574154741.781,
      "1"
    ],
    [
      1574154771.781,
```

```
    "1"
  ],
  [
    1574154801.781,
    "1"
  ],
  [
    1574154831.781,
    "1"
  ],
  [
    1574154861.781,
    "1"
  ],
  [
    1574154891.781,
    "1"
  ],
  [
    1574154921.781,
    "1"
  ],
  [
    1574154951.781,
    "1"
  ]
]
},
{
  "metric":{
    "__name__":"up",
    "instance":"192.168.0.7:9100",
    "job":"openstack"
  },
  "values":[
    [
      1574154651.781,
      "1"
    ],
    [
      1574154681.781,
      "1"
    ],
  ],
}
```

```
[
  [
    1574154711.781,
    "1"
  ],
  [
    1574154741.781,
    "1"
  ],
  [
    1574154771.781,
    "1"
  ],
  [
    1574154801.781,
    "1"
  ],
  [
    1574154831.781,
    "1"
  ],
  [
    1574154861.781,
    "1"
  ],
  [
    1574154891.781,
    "1"
  ],
  [
    1574154921.781,
    "1"
  ],
  [
    1574154951.781,
    "1"
  ]
]
},
{
  "metric":{
    "__name__":"up",
    "instance":"localhost:9090",
    "job":"prometheus"
  }
}
```

```
},  
"values":  
  [  
    1574154651.781,  
    "1"  
  ],  
  [  
    1574154681.781,  
    "1"  
  ],  
  [  
    1574154711.781,  
    "1"  
  ],  
  [  
    1574154741.781,  
    "1"  
  ],  
  [  
    1574154771.781,  
    "1"  
  ],  
  [  
    1574154801.781,  
    "1"  
  ],  
  [  
    1574154831.781,  
    "1"  
  ],  
  [  
    1574154861.781,  
    "1"  
  ],  
  [  
    1574154891.781,  
    "1"  
  ],  
  [  
    1574154921.781,  
    "1"  
  ],  
  [  
    ]
```

1574154951.781,

"1"

]

]

}

]

}

}