

CheckForUpdates

`/osc/checkForUpdates` API 通过比较客户端上一个已知的 `stateFingerprint` 和相机当前的 `fingerprint` 标识状态更新。

输入

名称	类型	说明
<code>stateFingerprint</code>	字符串	客户端上一次调用 <code>/osc/state</code> 或 <code>/osc/checkForUpdates</code> 的相机状态指纹。
<code>waitTimeout</code>	整型	在返回响应之前，等待相机状态发生变化的秒数。 <code>waitTimeout</code> 到期时，相机应返回一个响应，即使指纹未更改。如果在 <code>waitTimeout</code> 到期前检测到状态更改，或如果已忽略 <code>waitTimeout</code> ，则相机应立即返回响应。注：相机可以在 <code>waitTimeout</code> 到期之前返回响应，即使指纹未更改，但最佳做法是等待 <code>waitTimeout</code> 到期。

相机实现备注：

- 收到此调用时，相机将它的当前状态指纹与收到的 `stateFingerprint` 参数进行比较。如果指纹已更改，相机必须立即返回新的指纹。

输出

名称	类型	说明
<code>stateFingerprint</code>	字符串	相机状态的新指纹（与 <code>/osc/state</code> API 中的相同）。
<code>throttleTimeout</code>	整型	在下次调用 <code>checkForUpdates</code> 之前建议的客户端等待秒数。客户端可以在 <code>throttleTimeout</code> 到期前进行请求，如果可以的话，相机应允许这些提前的请求。

客户端实现备注：

- 收到响应后，客户端应比较收到的 `stateFingerprint` 及其副本。如果不匹配，客户端应使用 `_/osc/state` API 请求相机的当前状态。
- 智能客户端将阻止请求，无论相机如何响应。例如，如果相机返回一个非标准响应（即时，`_`没有变化且 `throttleTimeout`）较低或为 0，则客户端在通过相机请求另一个 `checkForUpdates` 之前应对自己的 `throttleTimeout` 实施限制。

相机实现备注：

- 在响应 `checkForUpdates` 时，相机应确定一个合理的 `throttleTimeout`。如果相机支持长时间的请求逻辑（仅在状态未更改时才会在 `waitTimeout` 后响应），则 `throttleTimeout` 的返回值为 0 是正常的。在此情况下，客户端可以立即请求更新。
- 如果相机仅支持快速响应（*不推荐*），则应返回一个合理的 `throttleTimeout`，以避免持续的请求/响应流向客户端。例如，合理的 `throttleTimeout` 为 60 秒，以允许每分钟处理一个客户端请求。
- **最佳做法**是返回适合相机功能的 `throttleTimeout`。如果服务器由于服务器问题而无法确定适当的 `throttleTimeout`，则相机应以 5XX 状态代码和一个包含 `serverError` 错误代码的 JSON 正文进行响应。

错误

错误代码	说明
<code>missingParameter</code>	未指定 <code>stateFingerprint</code> 。
<code>invalidParameterName</code>	一个或多个输入参数名称未被识别。
<code>invalidParameterValue</code>	参数名称已被识别，但一个或多个值无效；例如， <code>waitTimeout</code> 超出范围或其类型不正确。
<code>serverError</code>	服务器无法针对其响应确定适合的 <code>throttleTimeout</code> 值。服务器问题由响应中返回的 5XX 值标识。相机制造商应提供一个 5XX 代码表，以及会引发此错误的相应服务器状态。

示例

示例

请求

```
POST /osc/checkForUpdates HTTP/1.1
Host: [camera ip address]:[httpUpdatesPort]
Content-Type: application/json;charset=utf-8
Accept: application/json
Content-Length: {CONTENT_LEN}
X-XSRF-Protected: 1

{
  "stateFingerprint": "12EGA33",
  "waitTimeout": 300
}
```

响应

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=utf-8
Content-Length: {CONTENT_LENGTH}
X-Content-Type-Options: nosniff

{
  "stateFingerprint": "12EGA86",
  "throttleTimeout": 60
}
```

请求

```
POST /osc/checkForUpdates HTTP/1.1
Host: [camera ip address]:[httpUpdatesPort]
Content-Type: application/json;charset=utf-8
Accept: application/json
Content-Length: {CONTENT_LEN}
X-XSRF-Protected: 1

{
  "stateFingerprint": "12EGA33",
  "waitTimeout": 300
}
```

响应

```
HTTP/1.1 400 Bad Request
Content-Type: application/json;charset=utf-8
Content-Length: {CONTENT_LENGTH}
X-Content-Type-Options: nosniff

{
  "name": "camera.checkForUpdates",
  "state": "error",
  "error": {
    "code": "missingParameter",
    "message": "parameter stateFingerprint is mi
  }
}
```

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](https://creativecommons.org/licenses/by/3.0/) (<https://creativecommons.org/licenses/by/3.0/>), and code samples are licensed under the [Apache 2.0 License](https://www.apache.org/licenses/LICENSE-2.0) (<https://www.apache.org/licenses/LICENSE-2.0>). For details, see our [Site Policies](https://developers.google.com/terms/site-policies?hl=zh-cn) (<https://developers.google.com/terms/site-policies?hl=zh-cn>). Java is a registered trademark of Oracle and/or its affiliates.

上次更新日期: 三月 23, 2016