

# Agentenkonfiguration anlegen

- 1. Host anlegen und/oder ID und UUID zu existierendem Host bereit halten
- 2. Daten des Host abrufen
- 3. Agent Services abrufen / auf konfigurierten Agent prüfen
- 4. Agentkonfiguration übermitteln
  - Agent im Pull-Modus
    - Standardkonfiguration abrufen
    - Konfiguration speichern
  - Agent im Push-Modus
- 5. Agent Services abrufen
- 6. Agent Services speichern

## 1. Host anlegen und/oder ID und UUID zu existierendem Host bereit halten

Siehe [Hosts \(API\)](#)

## 2. Daten des Host abrufen

Dieser Schritt muss nur ausgeführt werden, wenn die UUID des Hosts noch nicht bekannt ist.

**URL:** <https://123.123.123.123/hosts/loadHostById/{HostID}.json?angular=true>

**Methode:** GET

**Antwort:** JSON

### Beispielantwort

```
{
  "host": {
    "id": 1,
    "uuid": "c36b8048-93ce-4385-ac19-ab5c90574b77",
    "container_id": 1,
    "name": "default host",
    "description": null,
    "hosttemplate_id": 1,
    "address": "localhost",
    "command_id": null,
    "eventhandler_command_id": null,
    "timeperiod_id": null,
    "check_interval": null,
    "retry_interval": null,
    "max_check_attempts": null,
    "first_notification_delay": null,
    "notification_interval": null,
    "notify_on_down": null,
    "notify_on_unreachable": null,
    "notify_on_recovery": null,
    "notify_on_flapping": null,
    "notify_on_downtime": null,
    "flap_detection_enabled": null,
    "flap_detection_on_up": null,
    "flap_detection_on_down": null,
    "flap_detection_on_unreachable": null,
    "low_flap_threshold": null,
  }
}
```

```

"high_flap_threshold": null,
"process_performance_data": null,
"freshness_checks_enabled": null,
"freshness_threshold": null,
"passive_checks_enabled": null,
"event_handler_enabled": null,
"active_checks_enabled": null,
"retain_status_information": null,
"retain_nonstatus_information": null,
"notifications_enabled": null,
"notes": null,
"priority": null,
"check_period_id": null,
"notify_period_id": null,
"tags": null,
"own_contacts": 0,
"own_contactgroups": 0,
"own_customvariables": 0,
"host_url": null,
"satellite_id": 0,
"host_type": 1,
"disabled": 0,
"usage_flag": 0,
"created": "2015-01-15T19:26:32+01:00",
"modified": "2020-07-15T11:54:14+02:00",
"hosts_to_containers_sharing": [
  {
    "id": 1,
    "containertype_id": 1,
    "name": "root",
    "parent_id": null,
    "lft": 1,
    "rght": 20,
    "_joinData": {
      "id": 1,
      "host_id": 1,
      "container_id": 1
    }
  }
],
"is_satellite_host": false,
"allow_edit": true
},
"_csrfToken": null
}

```

### 3. Agent Services abrufen / auf konfigurierten Agent prüfen

**URL:** [https://123.123.123.123/agentconnector/getServicesToCreateByHostUuid/\\${HostUUID}.json](https://123.123.123.123/agentconnector/getServicesToCreateByHostUuid/${HostUUID}.json)

**Methode:** GET

**Antwort:** JSON

Folgendes ist die Standard-Rückgabe, wenn noch kein Agent für diesen Host konfiguriert wurde. Sie muss nicht gespeichert werden.

#### Beispielantwort

```
{
  "servicesToCreate": "",
  "mode": "",
  "config": "",
  "system": "windows",
  "error": "",
  "_csrfToken": null
}
```

Wenn noch kein Agent konfiguriert wurde, fahre fort mit [Schritt 4](#).

Wenn bereits ein Agent konfiguriert wurde, gehe zu [Agentenkonfiguration bearbeiten](#).

## 4. Agentkonfiguration übermitteln

Entscheide, ob der Agent im Pull- oder Push-Modus angelegt werden soll.

### Agent im Pull-Modus

#### Standardkonfiguration abrufen

**URL:** `https://123.123.123.123/agentconfigs/config/${HostID}.json?angular=true`

**Methode:** GET

**Antwort:** JSON

## Beispielantwort

```
{
  "host": {
    "id": 1,
    "uuid": "c36b8048-93ce-4385-ac19-ab5c90574b77",
    "name": "default host",
    "address": "localhost",
    "container_id": 1,
    "hosts_to_containers_sharing": [
      {
        "id": 1,
        "containertype_id": 1,
        "name": "root",
        "parent_id": null,
        "lft": 1,
        "rght": 20,
        "_joinData": {
          "id": 1,
          "host_id": 1,
          "container_id": 1
        }
      }
    ]
  },
  "config": {
    "port": 3333,
    "use_https": 0,
    "insecure": 1,
    "basic_auth": 0,
    "proxy": 1,
    "username": "",
    "password": "",
    "push_noticed": 0
  },
  "_csrfToken": null
}
```

## Konfiguration speichern

**URL:** <https://123.123.123.123/agentconfigs/add.json?angular=true>

**Methode:** POST

**Content-Type:** application/json

**Antwort:** JSON

#### Beispiel Anfrageninhalt

```
{
  "Agentconfig": {
    "host_id": 1,
    "port": 3333,
    "use_https": false,
    "basic_auth": 0,
    "username": "",
    "password": ""
  }
}
```

#### Beispielantwort

```
{
  "id": 5,
  "_csrfToken": null
}
```

Die neue Konfiguration in die Konfigurationsdatei des Agents einfügen und diesen neu starten.

## Agent im Push-Modus

Die neue Konfiguration in die Konfigurationsdatei des Agents einfügen und diesen neu starten.

## 5. Agent Services abrufen

**Bevor fortgefahren wird sollten Sie sicher sein, dass der Agent läuft und erreichbar ist!**

**URL:** [https://123.123.123.123/agentconnector/getServicesToCreateByHostUuid/\\${HostUUID}.json](https://123.123.123.123/agentconnector/getServicesToCreateByHostUuid/${HostUUID}.json)

**Methode:** GET

**Antwort:** JSON

#### Beispielantwort mit Fehler

```
{
  "servicesToCreate": "",
  "mode": "could-be-pull",
  "config": "",
  "system": "windows",
  "error": "cURL error 7: Failed to connect to localhost port 3333: Connection refused (see https://\curl.haxx.se/libcurl/c/libcurl-errors.html)",
  "_csrfToken": null
}
```

Wenn wie in oben stehender Beispielantwort ein Fehler auftaucht, muss dieser nicht endgültig sein.

Führen Sie bei einem Fehler zur Sicherheit noch 2-3 weitere Anfragen aus. openITCOCKPIT wird versuchen den Agent mit/ohne SSL und mit/ohne Proxy zu erreichen.

Sollte dieser oder ein anderer Fehler auch nach 4 Anfragen nicht verschwinden, gibt es womöglich einen Fehler in der Konfiguration des Agents oder dieser kann nicht erreicht werden.

Der Wert im Feld "mode" weist auf die erkannte Agentkonfiguration hin. Auch wenn Fehler aufgetreten sind, ist es relativ sicher, dass es eine Konfiguration des Agents im "mode" Modus gibt. (Mögliche Werte bei Fehlern: "could-be-pull" oder "could-be-push")

Sollte weder in "error" noch in "servicesToCreate" etwas stehen, bzw. der Agent im Push Modus konfiguriert sein, sollte bis zu 2 Minuten lang alle 10 Sekunden eine Anfrage getätigt werden, da gewartet werden muss, bis der Agent erste Ergebnisse ans openITCOCKPIT sendet.

Um fortzufahren muss "error" in der Antwort leer und "servicesToCreate" gefüllt sein!

#### Beispielantwort ohne Fehler

```
{
  "servicesToCreate": {
    "Agent": [
      {
        "name": "OITC_AGENT_ACTIVE",
        "servicetemplate_id": 50
      }
    ],
    "SystemLoad": [
      {
        "name": "System Load",
        "servicetemplate_id": 52
      }
    ],
    "MemoryUsage": [
      {
        "name": "Memory Usage",
        "servicetemplate_id": 53
      }
    ],
    "SwapUsage": [
      {
        "name": "SWAP Usage",
        "servicetemplate_id": 54
      }
    ],
    "DiskUsage": [
      {
        "name": "Disk Usage /",
        "servicetemplate_id": 56,
        "agent_wizard_option_description": "/"
      },
      {
        "name": "Disk Usage /rpool",
        "servicetemplate_id": 56,
        "agent_wizard_option_description": "/rpool"
      }
    ],
    "DiskIO": [
      {
        "name": "Disk Load loop0",
        "servicetemplate_id": 55,
        "agent_wizard_option_description": "loop0"
      }
    ],
  }
}
```

```
{
  "name": "Disk Load sdb",
  "servicetemplate_id": 55,
  "agent_wizard_option_description": "sdb"
},
"NetStats": [
  {
    "name": "Network Device Status vmbr0",
    "servicetemplate_id": 61,
    "agent_wizard_option_description": "vmbr0"
  },
  {
    "name": "Network Device Status eth0",
    "servicetemplate_id": 61,
    "agent_wizard_option_description": "eth0"
  }
],
"NetIO": [
  {
    "name": "Network Stats vmbr0",
    "servicetemplate_id": 60,
    "agent_wizard_option_description": "vmbr0"
  },
  {
    "name": "Network Stats eth0",
    "servicetemplate_id": 60,
    "agent_wizard_option_description": "eth0"
  }
],
"Temperature": [
  {
    "name": "Device Temperature coretemp",
    "servicetemplate_id": 58,
    "agent_wizard_option_description": "coretemp"
  }
],
"CpuTotalPercentage": [
  {
    "name": "Total CPU Percentage",
    "servicetemplate_id": 51
  }
],
"Process": [
  {
    "name": "Check Process /sbin/init",
    "servicetemplate_id": 62,
    "agent_wizard_option_description": "/sbin/init"
  },
  {
    "name": "Check Process /lib/systemd/systemd --user",
    "servicetemplate_id": 62,
    "agent_wizard_option_description": "/lib/systemd/systemd --user"
  },
  {
    "name": "Check Process /lib/systemd/systemd-logind",
    "servicetemplate_id": 62,
    "agent_wizard_option_description": "/lib/systemd/systemd-logind"
  }
],
"SystemService": [
  {
    "name": "Check systemd service running apache2.service",
```

```
        "servicetemplate_id": 64,
        "agent_wizard_option_description": "apache2.service"
    },
    {
        "name": "Check systemd service running apparmor.service",
        "servicetemplate_id": 64,
        "agent_wizard_option_description": "apparmor.service"
    }
]
},
"mode": "pull",
"config": "",
"system": "linux",
"error": "",
"_csrfToken": null
}
```

Der Wert im Feld "mode" weist auf die erkannte Agentkonfiguration hin. (Mögliche Werte ohne Fehler: "pull" oder "push")

Der Wert im Feld "system" spezifiziert das Hostsystem des Agents. Je nach System unterscheiden sich die Pfade im Dateisystem zur Agentkonfiguration. (Mögliche Antworten: "windows", "linux", "macos")

In "servicesToCreate" sind die zur Verfügung stehenden Checks nach Gruppen sortiert (z.B. SystemdServices oder DiskUsage).

## 6. Agent Services speichern

Checks für die Services angelegt werden sollen, müssen aus "servicesToCreate" extrahiert werden und in einer neuen Post Anfrage im Array "serviceConfigs" übermittelt werden.

Die Gruppierungen (wie z.B. SystemdServices oder DiskUsage) können dabei ignoriert werden.

Das Feld "hostId" enthält die bereits zuvor genutzte Host ID.

Dem Feld "tryAutosslInPullMode" sollte die Zeichenkette "false" zugeordnet werden. Wenn AutoSSL zuvor aktiviert wurde, sollte hier natürlich "true" angegeben werden.

**URL:** <https://123.123.123.123/agentconnector/createServices.json?angular=true>

**Methode:** POST

**Content-Type:** application/json

**Antwort:** JSON



## Beispiel Anfrageninhalt

```
{
  "serviceConfigs": [
    {
      "name": "Total CPU Percentage",
      "servicetemplate_id": 51
    },
    {
      "name": "System Load",
      "servicetemplate_id": 52
    },
    {
      "name": "Memory Usage",
      "servicetemplate_id": 53
    },
    {
      "name": "SWAP Usage",
      "servicetemplate_id": 54
    },
    {
      "name": "Disk Usage /",
      "servicetemplate_id": 56,
      "agent_wizard_option_description": "/"
    },
    {
      "name": "Disk Load sdb",
      "servicetemplate_id": 55,
      "agent_wizard_option_description": "sdb"
    },
    {
      "name": "Device Temperature coretemp",
      "servicetemplate_id": 58,
      "agent_wizard_option_description": "coretemp"
    },
    {
      "name": "Network Stats eth0",
      "servicetemplate_id": 60,
      "agent_wizard_option_description": "eth0"
    },
    {
      "name": "Network Device Status eth0",
      "servicetemplate_id": 61,
      "agent_wizard_option_description": "eth0"
    },
    {
      "name": "Check Process /sbin/init",
      "servicetemplate_id": 62,
      "agent_wizard_option_description": "/sbin/init"
    },
    {
      "name": "Check systemd service running apache2.service",
      "servicetemplate_id": 64,
      "agent_wizard_option_description": "apache2.service"
    }
  ],
  "hostId": 6,
  "tryAutosslInPullMode": false
}
```

**Beispielantwort**

```
{  
  "success": "true",  
  "_csrfToken": null  
}
```