beh.cam

HTTP REST API Dokumentation

beh.digital GmbH

2022-08-11

Inhaltsverzeichnis

Vorwort

Einführung

Die beh.digital GmbH hat dieses Handbuch gewissenhaft erstellt. Dennoch kann keine Gewähr für den Inhalt, die Vollständigkeit und die Qualität der Informationen in dessen erfolgen. In regelmäßigen

Abständen und bei Updates wird dieses Handbuch aktualisiert und überprüft. Bei fehlenden Informationen oder Problemen mit unserem Produkt wenden sie sich bitte an ihren Ansprechpartner oder schreiben sie uns eine E-Mail an: support@beh.digital.

Auch beim Befolgen der in diesem Handbuch enthaltenden Bedienungsanleitung können wir nicht garantieren, dass es nicht zu Problem der Hardware und Software kommen kann. Des Weiteren können wir mit dem Kauf der Kamera nicht versprechen, dass das gewünschte Ergebnis erzielt wird. Soweit gesetzlich zulässig, ist die Haftung für direkte Schäden, Folgeschäden und Schäden Dritter, die sich aus dem Kauf dieses Produkts ergeben, ausgeschlossen. Die Haftung beschränkt sich auf den im Kaufvertrag angegebenen Produktpreis. Alle Verpflichtungen und Gewährleistungsregelungen sind ausschließlich in dem Kaufvertrag enthalten. Demnach stellt dieses Handbuch weder eine Erweiterung noch eine Einschränkung dar. Durch den Kauf wird dem Käufer das Recht zur Nutzung der Software eingeräumt.

Copyright

Kontakt & Support

Sie erreichen uns unter


```
beh.digital GmbH
Hespenkamp 1
DE-30419 Hannover
+49 (0)511 10582557
info@beh.digital
https://beh.digital
```


Versionshistorie

Version	Datum	Verantwortlicher	Bemerkung
1.0.0	11.08.2022	Behmann	Initiale Veröffentlichung

HTTP REST Schnittstelle

Die beh.cam stellt eine HTTP REST-basierte API zur Verfügung. Diese wird ebenfalls von der integrierten Weboberfläche verwendet. Die Schnittstelle verfügt über keine internen Zustände, die die Ausgabe einer Anfrage beeinflusst.

- Etablierte Englisch-sprachige Fachbegriffe wurden in der nachfolgenden, Endpunkt-spezifischen Dokumentation übernommen und nicht ins Deutsche übersetzt.
- Exemplarische Rückgaben der Schnittstelle wurden zur besseren Lesbarkeit mit Zeilenumbrüchen dargestellt, die in der rohen Rückgabe nicht enthalten sind.
- Falls eine Header Regexp angegeben ist, muss diese erfüllt sein, um eine Ausgabe zu erhalten. Anderenfalls wird der Fehlercode 404 (nicht gefunden) zurückgegeben.

Spezifische Datentypen

Tasktyp Datentyp

Der *type* bei der Rückgabe eines Task kennzeichnet das verwendete Template:

type	Task-Type	Beschreibung
0	Klassifikation	Klassifikation eines Bild(ausschnitts) in eine trainierte Klasse
1	Detektion	Erkennung (Lokalisierung und Klassifikation) mehrerer Objekte innerhalb eines Bildes
2	Multi Task	Ausführung mehrerer Klassifikations-Modelle an festen Positionen
3	*reserviert*	*reserviert für zukünftige Anomalie-Erkennung*
4	Custom Task	Ausführung von Bildverarbeitung auf der Kamera

Task-spezifische Endpunkte

Get Tasks

Gibt die auf der Kamera gespeicherten Tasks / Aufgaben als JSON-Array zurück.

Endpunkte: `/api/tasks`

Methoden: `GET`

Headers Regexp: `{"Accept": "application/json"}`

Exemplarische Rückgabe:


```
[
    {
        "id": "c9on9kjp8j5s5onmtbgg",
        "name": "Zahnraderkennung",
        "type": 0,
        "classes": [
            {
                "id": 2,
                "name": "Belegt",
                "profinetID": 1,
                "gpio0": 1,
                "gpio1": 0
            },
            {
                "id": 3,
                "name": "Leer",
                "profinetID": 0,
                "gpio0": 0,
                "gpio1": 1
            }
        ],
        "createdAt": "2022-05-03T20:30:42.373304+02:00",
        "deletedAt": "0001-01-01T00:00:00Z"
    },
    {
        "id": "c8kugu75upvevhdund4g",
        "name": "Leerer Task",
        "type": 0,
        "createdAt": "2022-03-10T13:00:56.877834+01:00",
        "deletedAt": "0001-01-01T00:00:00Z"
    }
]
```


Query Parameter: - `type`: auf einen oder mehrere Task-Typen filtern. Standard: Alle Tasktypen werden ausgegeben - `deleted`: wenn `1`, werden bereits gelöschte Tasks zurückgegeben. Standard: Ausgabe nur noch nicht gelöschter Tasks - `limit`: maximale Anzahl an Tasks in der Ausgabe. Standard: 30 -

`offset`: Start der begrenzten Ausgabe an Tasks. Standard: 0

Add Task

Hinzufügen eines Tasks.

Endpunkte: `/api/tasks`

Methoden: `POST`

Headers Regexp: `{"Accept": "application/json"}`

Dokumentation folgt

Task Import (zip)

Importiert einen Task aus einem ZIP. Dieses kann beispielsweise über `GET Task (as zip)` (auf einem anderen Gerät) exportiert worden sein.

Endpunkte: `/api/tasks/import`

Methoden: `POST`

Headers Regexp: `{"Content-Type": "application/zip"}`

Available Task Templates

Gibt die zur Verfügung stehenden Templates der Kamera zurück. Array der unterstützten Tasktypes.

Endpunkte: `/api/tasks/templates`

Methoden: `GET`

Headers Regexp: `{"Content-Type": "application/json"}`

Exemplarische Rückgabe:

```
[
    0,
    2,
    4
]
```

Get Task

Gibt die Eigenschaften eines Tasks zurück. Format identisch zu einem Element der Rückgabe aus `GET Tasks`.

Endpunkte: `/api/tasks/{taskID:[0-9a-v]{20}}`

Methoden: `GET`

Headers Regexp: `{"Content-Type": "application/json"}`

Exemplarische Rückgabe:


```
{
    "id": "c9on9kjp8j5s5onmtbgg",
    "name": "Zahnraderkennung",
    "type": 0,
    "classes": [
        {
            "id": 2,
            "name": "Belegt",
            "profinetID": 1,
            "gpio0": 1,
            "gpio1": 0
        },
        {
            "id": 3,
            "name": "Leer",
            "profinetID": 0,
            "gpio0": 0,
            "gpio1": 0
        }
    ],
    "createdAt": "2022-05-03T20:30:42.373304+02:00",
    "deletedAt": "0001-01-01T00:00:00Z"
}
```


Export Task (zip)

Exportiert einen Task als Zip-Datei, beispielsweise als Backup, oder zur Vervielfältigung des Tasks auf mehrere Endgeräte.

Endpunkte: `/api/tasks/{taskID:[0-9a-v]{20}}`, `/api/tasks/{taskID:[0-9a-v]{20}}/task.zip`

Methoden: `GET`

Headers Regexp: `{"Accept": "application/zip"}, {}`

Exemplarische Rückgabe: ZIP-Datei des Tasks, bestehend aus Bildern, Modellen und gespeicherten Samples sowie Informationen zum Task.

Delete Task

Löscht einen Task. Führt zunächst ein Soft-Delete (setzen des Feldes `deletedAt`) aus. Falls bereits Soft-Deleted, wird der Task hart gelöscht.

Endpunkte: `/api/tasks/{taskID:[0-9a-v]{20}}`

Methoden: `DELETE`

Rückgaben:

Code	Status	Bedeutung
204	No Content	Erfolgreich gelöscht
400	Bad Request	Task-ID unbekannt oder anderer Fehler

Update Task

Endpunkte: `/api/tasks/{taskID:[0-9a-v]{20}}`

Methoden: `PUT`

Dokumentation folgt

Patch Task

Endpunkte: `/api/tasks/{taskID:[0-9a-v]{20}}`

Methoden: `PATCH`

Dokumentation folgt

Task Image (first sample)

Gibt das Bild des ersten Samples aus einem Task zurück.

Endpunkte: `/api/tasks/{taskID:[0-9a-v]{20}}/image`

Methoden: `GET`

Ausgabe: PNG

Add Class to Task

Endpunkte: `/api/tasks/{taskID:[0-9a-v]{20}}/classes`

Methoden: `POST`

Dokumentation folgt

Delete Class from Task

Endpunkte: `/api/tasks/{taskID:[0-9a-v]{20}}/classes/{classID:[0-9]}`

Methoden: `DELETE`

Dokumentation folgt

Sample-spezifische Endpunkte

Get Samples

Gibt die Samples eines Tasks zurück.

Endpunkte: `/api/tasks/{taskID:[0-9a-v]{20}}/samples`

Methoden: `GET`

Headers Regexp: `{"Content-Type": "application/json"}`

Exemplarische Rückgabe:

```
[
    {
        "id": "brk8grf5upv9q48s522g",
        "taskID": "brk8d875upv9q48s51og",
        "image": "/api/tasks/brk8d875upv9q48s51og/samples/
           brk8grf5upv9q48s522g/image",
        "crop": {
            "x": 0,
```

```
            "y": 0,
            "w": 1,
            "h": 1
        },
        "validation": false,
        "createdAt": "2020-06-16T10:53:01.036+02:00",
        "deletedAt": "0001-01-01T00:00:00Z",
        "classID": 1
    },
    {
        "id": "brk8dvv5upv9q48s51sg",
        "taskID": "brk8d875upv9q48s51og",
        "image": "/api/tasks/brk8d875upv9q48s51og/samples/
           brk8dvv5upv9q48s51sg/image",
        "crop": {
            "x": 0,
            "y": 0,
            "w": 1,
            "h": 1
        },
        "validation": true,
        "createdAt": "2020-06-16T10:46:54.924+02:00",
        "deletedAt": "0001-01-01T00:00:00Z",
        "classID": 2
    },
    ...
]
```

Query Parameter: - `class_id`: auf einen oder mehrere Klassen-IDs filtern. Ausschließlich bei Klassifikations-Modellen Standard: Alle Klassen-IDs werden berücksichtigt - `validation`: wenn gesetzt, werden nur Bilder mit validation **`true`** (`1`) oder **`false`** (`0`) zurückgegeben. Standard: `0` und `1` - `deleted`: wenn `1`, werden bereits gelöschte Samples zurückgegeben. Standard: Ausgabe nur noch nicht gelöschter Samples - `limit`: maximale Anzahl an Tasks in der Ausgabe. Standard: 30 - `offset`: Start der begrenzten Ausgabe an Tasks. Standard: 0 - `random`: Zahl zwischen 0 und 1, mit welcher ein Seed der Sortierung vorgegeben werden kann. Standard: Sortierung nach ID

Add Sample

Fügt ein Sample dem entsprechenden Task hinzu.

Endpunkte: `/api/tasks/{taskID:[0-9a-v]{20}}/samples`

Methoden: `POST`

Dokumentation folgt

Capture Image as new Sample

Nimmt ein Bild auf und fügt dieses als neues (unklassifiziertes) Sample dem entsprechenden Task hinzu. Gibt dieses Sample zurück.

Endpunkte: `/api/tasks/{taskID:[0-9a-v]{20}}/capture`

Methoden: `GET`

Headers Regexp: `{"Content-Type": "application/json"}`

Dokumentation folgt

Get Sample

Gibt das Sample mit entsprechender ID zurück.

Endpunkte: `/api/tasks/{taskID:[0-9a-v]{20}}/samples/{sampleID:[0-9a-v]{20}}`

Methoden: `GET`

Headers Regexp: `{"Content-Type": "application/json"}`

Exemplarische Rückgabe:

```
{
    "id": "c9s1qbf5upvbepg4aicg",
    "taskID": "c9s1f075upva1tta4b40",
    "type": 4,
    "image": "/api/tasks/c9s1f075upva1tta4b40/samples/
       c9s1qbf5upvbepg4aicg/image",
    "crop": {
        "x": 0,
        "y": 0,
        "w": 1,
        "h": 1
    },
    "validation": false,
    "createdAt": "2022-05-08T21:42:37.316948+02:00",
    "deletedAt": "0001-01-01T00:00:00Z"
}
```

Delete Sample

Löscht das Sample aus dem Task. Führt zunächst ein Soft-Delete (setzen des Feldes `deletedAt`) aus. Falls bereits Soft-Deleted, wird das Sample hart gelöscht.

Endpunkte: `/api/tasks/{taskID:[0-9a-v]{20}}/samples/{sampleID:[0-9a-v]{20}}`

Methoden: `DELETE`

Rückgaben:

Code	Status	Bedeutung
204	No Content	Erfolgreich gelöscht
400	Bad Request	Task-ID unbekannt oder anderer Fehler

Patch Sample

Endpunkte: `/api/tasks/{taskID:[0-9a-v]{20}}/samples/{sampleID:[0-9a-v]{20}|all}`

Methoden: `PATCH`

Dokumentation folgt

Get Image of Sample

Gibt das Bild des Samples zurück.

Endpunkte: `/api/tasks/{taskID:[0-9a-v]{20}}/samples/{sampleID:[0-9a-v]{20}}/image`

Methoden: `GET`

Rückgabe: PNG

Model-spezifische Endpunkte

Inferenz-Ergebnis Struktur

Beispielhafte Ausgabe

```
{
    "modelID": "c0mh5hv5upvdn1dnhsl0",
    "type": 1,
    "roi": {
        "x": 0,
        "y": 0,
        "w": 1,
        "h": 1
    },
    "results": [
        {
            "classID": 1,
            "mappedClassID": 0,
            "className": "Objekt A",
            "confidence": 0.3515625,
            "bbox": {
                "x": 0.1083346,
                "y": 0.25184333,
                "w": 0.73278534,
                "h": 0.7098198
            }
        },
        {
            ...
        }
    ]
}
```

Eintrag	Datentyp	Beschreibung
modelID	string	ID des ausgeführten Models
type	Tasktyp	Art des ausgeführten Models
roi	Rectangle	Bildausschnitt, der betrachtet wurde
results	Inferenz-Ergebnis	Klassifikationsergebnis / Detektierte Objekte / Ausgabe

Get Models

Endpunkte: `/api/tasks/{taskID:[0-9a-v]{20}}/models`

Methoden: `GET`

Headers Regexp: `{"Content-Type": "application/json"}`

Exemplarische Rückgabe:


```
[
    {
        "id": "c9p500bp8j5sg3u9mu50",
        "type": 0,
        "taskID": "c9on9kjp8j5s5onmtbgg",
        "name": "Unnamed Model",
        "description": "No description",
        "createdAt": "2022-05-04T09:08:17.716+02:00",
        "deletedAt": "0001-01-01T00:00:00Z",
        "augmentationConfig": {
            "illuminationLower": 2,
            "illuminationLowerStep": 1,
            "illuminationUpper": 2,
            "illuminationUpperStep": 1,
            "grayscale": false,
            "flipHorizontally": true,
            "flipVertically": true,
            "flipBoth": true,
            "translationLeft": 0.1,
            "translationLeftStep": 0.1,
            "translationRight": 0.1,
            "translationRightStep": 0.1,
            "translationTop": 0.1,
            "translationTopStep": 0.1,
            "translationBottom": 0.1,
            "translationBottomStep": 0.1,
            "rotation": 0,
            "rotationStep": 0,
            "zoomOut": 1.2,
            "zoomOutStep": 0.2,
            "zoomIn": 1.2,
            "zoomInStep": 0.2
        },
        "trainConfig": {
            "classes": [
                {
                    "id": 2,
                    "name": "Belegt",
```

```
                    "profinetID": 1,
                    "gpio0": 1,
                    "gpio1": 0
                },
                {
                    "id": 3,
                    "name": "Leer",
                    "profinetID": 0,
                    "gpio0": 0,
                    "gpio1": 1
                }
            ],
            "name": "mobilenet_v2",
            "inputSize": 224,
            "depthMultiplier": 0.35,
            "debug": false,
            "hiddenUnits": 0,
            "learningRate": 0.005,
            "epochs": 100,
            "batchSize": 100,
            "validationSplit": -0.1,
            "batchLogs": "[{\"batch\":0,\"size\":39,\"loss
                \":3.109633445739746,\"acc\":0.07692307978868484},{...}]
                ",
            "epochLogs": "[{\"val_loss\":2.3543283939361572,\"val_acc
                \":0.25641027092933655,\"loss\":3.109633445739746,\"acc
                \":0.07692307978868484},{...}]"
        },
        "inferenceConfig": {
            "roi": {
                "x": 0,
                "y": 0,
                "w": 1,
                "h": 1
            },
            "gpio0func": 0,
            "gpio1func": 0
        }
    },
    {
        ...
    }
]
```

Add Model

Fügt ein Model einem Task hinzu.

Endpunkte: `/api/tasks/{taskID:[0-9a-v]{20}}/models`

Methoden: `POST`

Headers Regexp: `{"Content-Type": "multipart/form-data"}`

Dokumentation folgt

Import Model (zip)

Importiert ein Model aus einer ZIP-Datei in einen Task.

Endpunkte: `/api/tasks/{taskID:[0-9a-v]{20}}/models/import`

Methoden: `POST`

Headers Regexp: `{"Content-Type": "application/zip"}`

Dokumentation folgt

Get Model

Gibt das Model mit angefragter Model-ID zurück. Die Task-ID kann entfallen.

Endpunkte: `/api/models/{modelID:[0-9a-v]{20}}`, `/api/tasks/{taskID:[0-9a-v]{20}}/models/{modelID:[0-9a-v]{20}}`

Methoden: `GET`

Headers Regexp: `{"Content-Type": "application/json"}`

Exemplarische Rückgabe:

```
{
    "id": "c9p500bp8j5sg3u9mu50",
    "type": 0,
    "taskID": "c9on9kjp8j5s5onmtbgg",
    "name": "Unnamed Model",
    "description": "No description",
    "createdAt": "2022-05-04T09:08:17.716+02:00",
```

```
    "deletedAt": "0001-01-01T00:00:00Z",
    "augmentationConfig": {
        "illuminationLower": 2,
        "illuminationLowerStep": 1,
        "illuminationUpper": 2,
        "illuminationUpperStep": 1,
        "grayscale": false,
        "flipHorizontally": true,
        "flipVertically": true,
        "flipBoth": true,
        "translationLeft": 0.1,
        "translationLeftStep": 0.1,
        "translationRight": 0.1,
        "translationRightStep": 0.1,
        "translationTop": 0.1,
        "translationTopStep": 0.1,
        "translationBottom": 0.1,
        "translationBottomStep": 0.1,
        "rotation": 0,
        "rotationStep": 0,
        "zoomOut": 1.2,
        "zoomOutStep": 0.2,
        "zoomIn": 1.2,
        "zoomInStep": 0.2
    },
    "trainConfig": {
        "classes": [
            {
                "id": 2,
                "name": "Belegt",
                "profinetID": 1,
                "gpio0": 1,
                "gpio1": 0
            },
            {
                "id": 3,
                "name": "Leer",
                "profinetID": 0,
                "gpio0": 0,
                "gpio1": 1
            }
        ],
        "name": "mobilenet_v2",
```

```
        "inputSize": 224,
        "depthMultiplier": 0.35,
        "debug": false,
        "hiddenUnits": 0,
        "learningRate": 0.005,
        "epochs": 100,
        "batchSize": 100,
        "validationSplit": -0.1,
        "batchLogs": "[{\"batch\":0,\"size\":39,\"loss
           \":3.109633445739746,\"acc\":0.07692307978868484},{...}]",
        "epochLogs": "[{\"val_loss\":2.3543283939361572,\"val_acc
           \":0.25641027092933655,\"loss\":3.109633445739746,\"acc
           \":0.07692307978868484},{...}]"
    },
    "inferenceConfig": {
        "roi": {
            "x": 0,
            "y": 0,
            "w": 1,
            "h": 1
        },
        "gpio0func": 0,
        "gpio1func": 0
    }
}
```

Patch Model

Modifizierte einzelne Felder eines Models.

Endpunkte: `/api/models/{modelID:[0-9a-v]{20}}`, `/api/tasks/{taskID:[0-9a-v]{20}}/models/{modelID:[0-9a-v]{20}}`

Methoden: `PATCH`

Headers Regexp: `{"Content-Type": "application/json"}`

Dokumentation folgt

Delete Model

Löscht das Model aus dem Task. Führt zunächst ein Soft-Delete (setzen des Feldes `deletedAt`) aus. Falls bereits Soft-Deleted, wird das Model hart gelöscht.

Endpunkte: `/api/models/{modelID:[0-9a-v]{20}}`, `/api/tasks/{taskID:[0-9a-v]{20}}/models/{modelID:[0-9a-v]{20}}`

Methoden: `DELETE`

Rückgaben:

Code	Status	Bedeutung
204	No Content	Erfolgreich gelöscht
400	Bad Request	Task-ID unbekannt oder anderer Fehler

Infer/Execute Model (single image)

Führt das Model einmalig auf dem aktuellen Bild aus (Inferenz).

Endpunkte: `/api/models/{modelID:[0-9a-v]{20}}/inference`

Methoden: `GET`

Headers Regexp: `{"Content-Type": "application/json"}`

Exemplarische Rückgabe:


```
{
    "modelID": "c0mh5hv5upvdn1dnhsl0",
    "type": 1,
    "roi": {
        "x": 0,
        "y": 0,
        "w": 1,
        "h": 1
    },
    "results": [
        {
            "classID": 1,
            "mappedClassID": 0,
            "className": "Objekt A",
            "confidence": 0.3515625,
            "bbox": {
                "x": 0.1083346,
                "y": 0.25184333,
                "w": 0.73278534,
                "h": 0.7098198
```

```
            }
        },
        {
            ...
        }
    ]
}
```

Infer/Execute Model (stream)

Startet die Ausführung des Models auf dem Videostrom. Es werden kontinuierlich Inferenz-Ergebnisse zurückgegeben. Die Rate der Inferenz entspricht dem Minimum aus Bildrate (30 Bilder pro Sekunde) und der Gesamtlatenz aller parallel ausgeführten Modelle. Die Anfrage muss über einen WebSocket gestartet werden (Upgrade).

Endpunkte: `/api/models/{modelID:[0-9a-v]{20}}/inference`, `/api/tasks/{taskID:[0-9a-v]{20}}/models/{modelID:[0-9a-v]{20}}`

Methoden: `GET`

Exemplarische Rückgabe (WebSocket-Message):

```
{
    "modelID": "c0mh5hv5upvdn1dnhsl0",
    "type": 1,
    "roi": {
        "x": 0,
        "y": 0,
        "w": 1,
        "h": 1
    },
    "results": [
        {
            "classID": 1,
            "mappedClassID": 0,
            "className": "Objekt A",
            "confidence": 0.3515625,
            "bbox": {
                "x": 0.1083346,
                "y": 0.25184333,
                "w": 0.73278534,
                "h": 0.7098198
            }
```

```
        },
        {
            ...
        }
    ]
}
```

Export Model (as zip)

Exportiert das Model als ZIP-Datei.

Endpunkte: `/api/tasks/{taskID:[0-9a-v]{20}}/models/{modelID:[0-9a-v]{20}}`, `/api/tasks/{taskID:[0-9a-v]{20}}/models/{modelID:[0-9a-v]{20}}/model.zip`

Methoden: `GET`

Headers Regexp: `{("Accept", "application/zip"),{}`

Get File of Model

Gibt eine Datei des Models zurück (z.B. `model.json`, `model.tflite`)

Endpunkte: `/api/tasks/{taskID:[0-9a-v]{20}}/models/{modelID:[0-9a-v]{20}}/{file}`

Methoden: `GET`

Update File of Model

Endpunkte: `/api/tasks/{taskID:[0-9a-v]{20}}/models/{modelID:[0-9a-v]{20}}/{file}`

Methoden: `PUT`

Dokumentation folgt

Video-spezifische Endpunkte

H264-Stream (live)

H264-enkodierter WebSocket Stream des Live Kamera-Bildes.

Endpunkte: `/api/video/stream/h264`

Methoden: `GET`

Video Einstellungen

Endpunkte: `/api/video/settings`

Methoden: `GET`

Video Einstellungen modifizieren

Endpunkte: `/api/video/settings`

Methoden: `PATCH`

Dokumentation folgt

Bild aufnehmen

Nimmt ein Bild auf und gibt es zurück.

Endpunkte: `/api/video/capture`

Methoden: `GET`

Rückgabe: PNG

Inferenz-spezifische Endpunkte

Listet die zur Zeit ausgeführten Modelle auf.

Endpunkte: `/api/inference/state`

Methoden: `GET`

Dokumentation folgt

Einstellungs-spezifische Endpunkte

GET Netzwerk-Einstellungen

Endpunkte: `/api/settings/network/{iface}`

Methoden: `GET`

Dokumentation folgt

ADD IP-Addresse zur Netzwerk-Schnittstelle

Endpunkte: `/api/settings/network/{iface}/ip`

Methoden: `POST`

Dokumentation folgt

DELETE IP-Addresse von Netzwerk-Schnittstelle

Endpunkte: `/api/settings/network/{iface}/ip`

Methoden: `DELETE`

Dokumentation folgt

SW-Update der Kamera

Endpunkte: `/api/settings/update`

Methoden: `POST`

Dokumentation folgt

GET Autostart Modelle

Gibt die Modelle zurück, welche beim Start (Boot) der Kamera automatisch gestartet werden.

Endpunkte: `/api/settings/models/autostart`

Methoden: `GET`

Headers Regexp: `{"Content-Type": "application/json"}`

Dokumentation folgt

Füge Autostart Modell hinzu

Endpunkte: `/api/settings/models/autostart`

Methoden: `POST`

Dokumentation folgt

DELETE Autostart Model

Endpunkte: `/api/settings/models/autostart/{modelID:[0-9a-v]{20}}`

Methoden: `DELETE`

Dokumentation folgt

GET Hardware Status

Beispielsweise Temperaturen der CPU.

Endpunkte: `/api/settings/hardware/status`

Methoden: `GET`

Headers Regexp: `{"Content-Type": "application/json"}`

Dokumentation folgt

Status-spezifische Endpunkte

GET Service Status

Endpunkte: `/api/service/{service}`

Methoden: `GET`, `POST`

Dokumentation folgt

RPC Service

Endpunkte: `/api/service/{service}/{command}`

Methoden: `GET`

Dokumentation folgt