Eye-Office API

Sessions/Authentifizierung

Für die Nutzung aller Endpunkt bis auf /v1/login und /v1/ping muss im HTTP-Header ein Feld namens `Session-ID` gesetzt sein. Diese Session-ID wird nach dem Aufruf von /v1/login zurückgegeben.

Außerdem erhält jedes Programm, das mit der Eye-Office API kommunizieren soll, einen API-Key zugewiesen. Dieser muss im HTTP-Header-Feld `X-API-KEY` übermittelt werden.

Paginierung

Einige Suchergebnisse werden in mehrere Seiten/Blöcke aufgeteilt (pagination). Die Dokumentation dieser Endpunkte enthält einen Verweis auf diesen Paginierung-Abschnitt.

Die gewünschte Blockgröße (Anzahl Datensätze pro Block) kann über den Parameter `pageSize` angegeben werden. Auf dem Server kann eine maximale Größe definiert sein, so dass die genutzte Blockgröße kleiner sein kann.

Die zurückgegebenen JSON-Daten enthalten zwei Elemente:

Das Element `data` enthält einen Block Datensätze. Das Element `pagination` enthält die Anzahl der Datensätze in diesem Block sowie einen Link (`linkNext`) zum Abrufen des nächsten Blocks. Wird statt eines Links `null` zurückgegeben, ist dies der letzte Datenblock. Vor dem Link muss die Basis-URL eingefügt werden.

Beispiel

```
{
  "data": [
    {
      ...
    },
    {
      ...
    }
  ],
  "pagination": {
    "pageSize": 1000,
    "linkNext": "/v1/customer?next=49213”
  }
}
```

Login

Jede Verbindung muss eine aktive Sitzung haben. Inaktive Sitzungen werden nach x Minuten (5?) wieder geschlossen.

Die zurückgegebene Session-ID muss bei jedem weiteren API-Aufruf mitgegeben werden.

POST /v1/ login

Media type

application/json

Request-Body

Beispiel:

```
{
   "user": "benutzer",
   "password": "password"
```

```
}
```

`user`: Ein für die API gültiger Benutzername

`password`: Das dazugehörige Passwort

Rückgabe

Code 200: Erfolgreich
Code 500: Internal Error (Datenbank-Fehler)

Rückgabe-Body

```
{
   "session": "session-id"
}
```

Logout

Damit nicht jede Sitzung durch ein Timeout geschlossen wird, muss der Logout-Endpunkt einmal mit der zugewiesenen Session-ID aufgerufen werden.

POST /v1/ logout

Rückgabe

Code 200: Erfolgreich
Code 400: Bad Request (z.B. keine oder eine ungültige Session-ID im Header)

Ping

Jede Sitzung wird nach Ablauf eines Timeouts automatisch geschlossen, wenn in der Zwischenzeit kein weiterer Request verschickt wurde. Ein Ping-Request setzt ebenfalls diesen Timer zurück.

Dieser Endpunkt funktioniert auch ohne gültige Session-ID.

POST /v1/ping

Rückgabe

Code 200: Erfolgreich

Kunden auflisten

Dieser Endpunkt sucht Kunden nach bestimmten Kriterien.

Die Ergebnisse sind paginiert, also in mehrere Seiten/Blöcke aufgeteilt (siehe oben).

GET /v1/customer

Optionale Parameter

- pageSize=2000
  Gibt pro API-Aufruf 2000 maximal Datensätze zurück. Fehlt dieser Parameter, wird eine konfigurierbare Standard-Größe verwendet.
- next=abc1234
  Gibt den nächsten Datenblock zurück. Dieser Parameter ist in Antwort-JSON-Struktur jeweils enthalten.
- lastChangedGreaterThan=2021-01-12T12:30:00
  Gibt Kundendaten zurück, die nach dem angegebenen Zeitstempel verändert wurden.
- firstname=georg
  Sucht nach Kunden mit dem Vornamen „Georg“.
- lastname=mustermann
  Sucht nach Kunden mit dem Nachnamen „Mustermann“.

- onlyActive=false
  Sollen nur aktive Kunden zurückgegeben werden? Standard: nur aktive Kunden.
- searchMode=contains
  Gibt an, ob der Suchbegriff irgendwo im entsprechenden Feld enthalten sein kann („contains“) oder am Anfang des Feldes stehen muss („startsWith“). Ist kein searchMode angegeben, wird „contains“ angenommen.

Media type

application/json

Rückgabe-Body

JSON-Struktur (Beispiel):

```
{
  "data": [
    {
      "id": 271,
      "firstname": "Georg",
      "lastname": "Mustermann",
      "birthday": "1970-01-01",
      "sex": "male",
      "title": "",
      "updatedAt": "2024-10-12T12:45:23",
      "isActive": true
    }
  ],
  "pagination": {
    "pageSize": 100,
    "linkNext": "/v1/customer?next=49213”
  }
}
```

Kundendaten zurückgeben

Mit diesem Endpunkt können Kundendaten abgefragt werden.

GET /v1/customer/{customerId}

Media type

application/json

Rückgabe

JSON-Struktur (Beispiel):

```
[
  {
    "id": 271,
    "firstname": "Georg",
    "lastname": "Mustermann",
    "birthday": "1970-01-01",
    "sex": "male",
    "title": "",
    "updatedAt": "2024-10-12T12:45:23",
    "isActive": true
  }
]
```

Mitarbeiter- und Ärzte

Um Refraktionen anzulegen werden die IDs von Mitarbeitern oder Ärzten benötigt. Eine Liste aller in Eye-Office angelegten Personen kann über diesen Endpunkt abgerufen werden.

GET /v1/people

Media type

application/json

Rückgabe

JSON-Struktur (Beispiel):

```
[
  {
    "id": 271,
    "type": "employee",
    "sex": "male",
    "title": "Dr.",
    "firstname": "Georg",
    "lastname": "Mustermann",
    "updatedAt": "2024-10-12T12:45:23"
  }
]
```

`type` enthält den Typen der Person, z.B. Mitarbeiter(`employee`) oder Arzt (`doctor`).

Refraktionsdaten zurückgeben

GET /v1/refraction

Obligatorische Parameter

- customerId=1234
  Gibt die Refraktionsdaten dieses Kunden zurück.

Optionale Parameter

- latestDataOnly=true
  Gibt nur die neuste Refraktion zurück (statt einer kompletten Liste). Default: false

Media type

application/json

Rückgabe

JSON-Struktur (Beispiel):

```
[
  {
    "id": 271,
    "useForOrder": "internal",
    "interalData": {
      "date": "2024-10-01",
      "person": {
        "id": 3456,
        "name": "Michael Müller"
      },
      "readingDistance": 40,
      "objectiveValues": {
        "rightEye": {
          "sphere": -1.0,
          "cylinder": null,
          "axisCylinder": null,
          "addition": 1.0,
          "prismHorizontalValue": null,
          "prismHorizontalAxis": null,
          "prismVerticalValue": null,
```

```
      "prismVerticalAxis": null,
      "backVertexDistance": 12,
      "visusCc": null
    },
    "leftEye": {
      "sphere": -2.5,
      "cylinder": null,
      "axisCylinder": null,
      "addition": 1.0,
      "prismHorizontalValue": null,
      "prismHorizontalAxis": null,
      "prismVerticalValue": null,
      "prismVerticalAxis": null,
      "backVertexDistance": 12,
      "visusCc": null
    },
    "visusCcBin": null
  },
  "subjectiveValues": {
    "rightEye": {
      "sphere": -1.0,
      "cylinder": null,
      "axisCylinder": null,
      "addition": 1.0,
      "prismHorizontalValue": null,
      "prismHorizontalAxis": null,
      "prismVerticalValue": null,
      "prismVerticalAxis": null,
      "backVertexDistance": 12,
      "interpupillaryDistance": 32,
      "visusSc": null
      "visusCc": null
    },
    "leftEye": {
      "sphere": -2.5,
      "cylinder": null,
      "axisCylinder": null,
      "addition": 1.0,
      "prismHorizontalValue": null,
      "prismHorizontalAxis": null,
      "prismVerticalValue": null,
      "prismVerticalAxis": null,
      "backVertexDistance": 12,
      "interpupillaryDistance": 32,
      "visusSc": null
      "visusCc": null
    },
    "visusScBin": null
    "visusCcBin": null
  },

},
"externalData": {
  "date": "2024-10-01",
  "person": {
    "id": 3456,
    "name": "Michael Müller"
  },
  "readingDistance": 40,
  "rightEye": {
    "sphere": -1.0,
    "cylinder": null,
    "axisCylinder": null,
```

```
        "addition": 1.0,
        "prismHorizontalValue": null,
        "prismHorizontalAxis": null,
        "prismVerticalValue": null,
        "prismVerticalAxis": null,
        "backVertexDistance": 12,
        "interpupillaryDistance": 32,
        "visusSc": null
        "visusCc": null
      },
      "leftEye": {
        "sphere": -2.5,
        "cylinder": null,
        "axisCylinder": null,
        "addition": 1.0,
        "prismHorizontalValue": null,
        "prismHorizontalAxis": null,
        "prismVerticalValue": null,
        "prismVerticalAxis": null,
        "backVertexDistance": 12,
        "interpupillaryDistance": 32,
        "visusSc": null
        "visusCc": null
      }
    },
    "comment": null
  }
]
```


`id` enthält die Datenbank-ID der Refraktion.

`useForOrder` gibt an, ob die Daten der Eigen- oder die Daten der Fremd-Refraktion für neue Aufträge genutzt werden soll.

`internalData` enthält die Daten der Eigenrefraktion.

`externalData` enthält die Daten der Fremdrefraktion.

`readingDistance` ist der Leseabstand in Zentimeter.

`backVertexDistance` ist der der Hornhaut-Scheitel-Abstand (HSA) in Millimeter.

`interpupillaryDistance` enthält den Mittenabstand, also die PD der Messbrille in Millimeter.

Prismen werden mit ihrem horizontalen und vertikalen Anteil, die Gradzahlen nach dem TABO-Gradbogenschema angegeben.

Grundsätzlich werden Felder, die nicht gesetzt sind, weggelassen. In diesem Beispiel sind sie auf `null` gesetzt.

Refraktion ergänzen

Speichert eine neue Refraktion zu einem Kunden ab. Die Datenstruktur entspricht der Struktur des entsprechenden GET-Endpunktes.

POST /v1/refraction

Media type

application/json

Request-Body

Beispiel:

```
{
  "useForOrder": "internal",
  "interalData": {
    "date": "2024-10-01",
    "personId": 3456,
    "readingDistance": 40,
...
  },
  "comment": null
}
```

Personen (Mitarbeiter oder Ärzte) werden über ihre ID referenziert. Diese kann über den Endpunkt GET /v1/people abgerufen werden.

Rückgabe

Code 200: Erfolgreich
Code 400: Bad Request (z.B. gar keine Refraktiondaten, Zylinder ohne Addition, …)
Code 500: Internal Error (Datenbank-Fehler)

Rückgabe-Body

```
{
   "id": 12345
}
```

`id` ist die Datenbank-ID der neu angelegten Refraktion.