HOCHSCHULE HEILBRONN
TECHNIK WIRTSCHAFT INFORMATIK

Software Engineering Bachelor

Embedded Systems

Projektplan

Heizungssteuerung mit ZigBee

Eugen Schlecht, Florens Hückstädt, Florian Wohlgemuth, Patrick Wohlgemuth

Inhaltsverzeichnis

1 Einführung

Im Zuge der Vorlesung Embedded Systems des Studiengangs Software Engineering Bachelor wird ein Projekt im Bereich der Heimautomatisierung durchgeführt. Thema des Projekts ist die Konzeption und Erstellung einer Heizungssteuerung. Dabei wird an einem im Sommersemester 2015 durchgeführten Projekt angeknüpft.

2 Ausgangssituation

3 Zieldefinition

4 Architektur und Umsetzung

Im Nachfolgenden werden die entwickelten und eingesetzten Lösungen zu den im Abschnitt 3 beschriebenen hardware- und softwarespezifischen Anforderungen erläutert. Abbildung 1 zeigt die grobe Systemarchitektur der entwickelten Lösung.

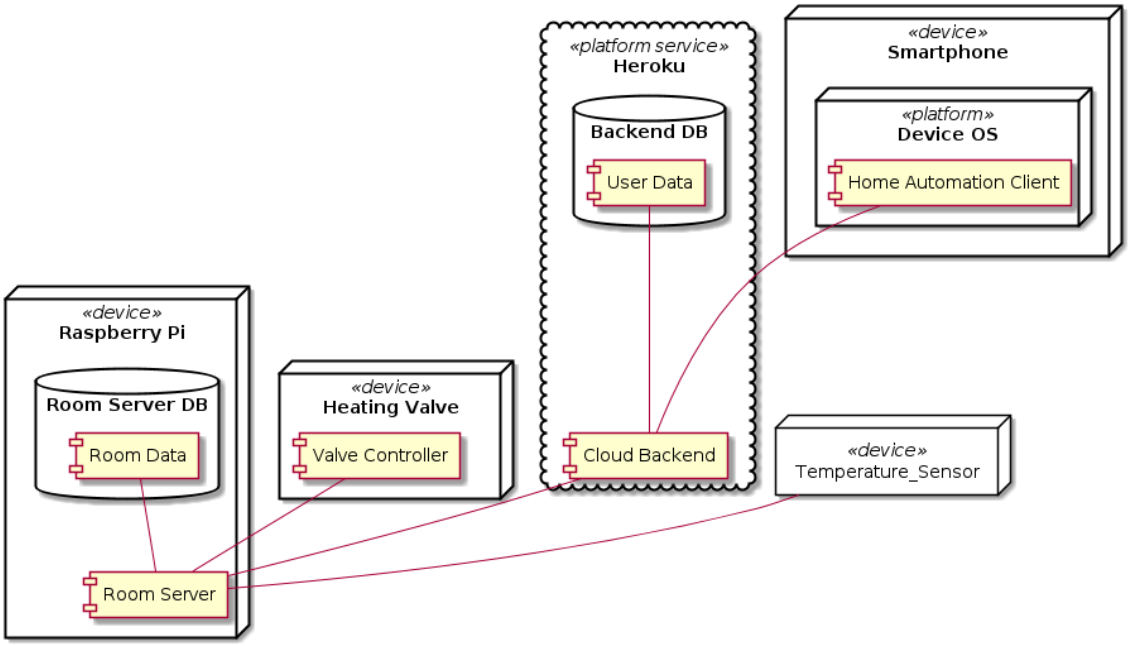

Abbildung 1: Sestemarchitektur

4.1 Hardware

Die Hardware

4.2 Software

Die Software

4.2.1 Backend

Das Backend kann über ein API von der App angesprochen werden, um Räume und Geräte zu verwalten und Kommandos an diese zu versenden.

API

Das API ist als RESTful Web Service realisiert. Ressourcen können über HTTP mit den Methoden `GET`, `POST`, `PUT` und `DELETE` angesprochen werden. Eine genaue Auflistung der API-Methoden ist im Anhang zu finden.

Schema

Sämtlicher Zugriff auf das API erfolgt über HTTPS und kann über `https://enigmatic-waters-31128.herokuapp.com` erreicht werden.

```
$ curl 'https://enigmatic-waters-31128.herokuapp.com/api/rooms/d934eb20-4c6f-4d1c-91c5-61cdeddcf843'
-i -H 'X-AUTH-TOKEN: TOKEN'

HTTP/1.1 200 OK
Connection: keep-alive
Server: Apache-Coyote/1.1
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: POST, PUT, PATCH, GET, OPTIONS, DELETE
Access-Control-Max-Age: 3600
Access-Control-Allow-Headers: x-requested-with, content-type, X-AUTH-TOKEN
Access-Control-Expose-Headers: X-AUTH-TOKEN
X-Content-Type-Options: nosniff
X-Xss-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Strict-Transport-Security: max-age=31536000 ; includeSubDomains
X-Frame-Options: DENY
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
Date: Sun, 01 May 2016 10:57:10 GMT
Via: 1.1 vegur

{
 "roomId":"d934eb20-4c6f-4d1c-91c5-61cdeddcf843",
 "name":"Wohnzimmer"
}
```

Sowohl beim Abfragen einer Liste als auch beim Abfragen einzelner Ressourcen beinhaltet die Antwort alle Attribute der Ressource.

Authentifizierung

Anfragen, die eine Authentifizierung erfordern, geben als Antwort `403 Forbidden` zurück. Zur Authentifizierung muss im Header das Feld `X-AUTH-TOKEN` gesetzt sein. Das entsprechende Token wird nach einer erfolgreichen Login-Anfrage erhalten.

Authentifizierung mit einem Token

```
$ curl 'https://enigmatic-waters-31128.herokuapp.com' -i -H 'X-AUTH-TOKEN: TOKEN'
```

Login

Zum Erhalt eines Tokens muss eine Loginanfrage gestellt werden. Diese enthält die Attribute username und password.

```
$ curl 'https://enigmatic-waters-31128.herokuapp.com/api/login' -i -X POST -H 'Content-Type: application/json
    ' -d '{"username":"foo", "password": "bar"}'

HTTP/1.1 200 OK
Connection: keep-alive
Server: Apache-Coyote/1.1
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: POST, PUT, PATCH, GET, OPTIONS, DELETE
Access-Control-Max-Age: 3600
Access-Control-Allow-Headers: x-requested-with, content-type, X-AUTH-TOKEN
Access-Control-Expose-Headers: X-AUTH-TOKEN
X-Content-Type-Options: nosniff
X-Xss-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Strict-Transport-Security: max-age=31536000 ; includeSubDomains
X-Frame-Options: DENY
X-Auth-Token: eyJpZCI6MTEsInVzZXJuYW1lIjoidXNlciIsImV4cGlyZXMiOjE0NjI5NjkONzIwMTgsInJvbGVzIjpbIlVTRVIiXX0=.
    LDd74GOyJAuQChYR9POAOmThfylOGGlY19u2DcTcXyQ=
Content-Length: 0
Date: Sun, 01 May 2016 12:24:32 GMT
Via: 1.1 vegur
```

Das Token aus dem Response-Header-Feld `X-AUTH-TOKEN` muss bei zukünftigen Anfragen enthalten sein.

Fehlgeschlagener Login

Loginanfragen mit ungültigen Benutzerdaten werden mit `401 Unauthorized` beantwortet.

```
$ curl 'https://enigmatic-waters-31128.herokuapp.com/api/login' -i -X POST -H 'Content-Type: application/json
    ' -d '{"username":"foo", "password": "bar"}'

HTTP/1.1 401 Unauthorized
Connection: keep-alive
Server: Apache-Coyote/1.1
Access-Control-Allow-Origin: *
Access-Control-Allow-Methods: POST, PUT, PATCH, GET, OPTIONS, DELETE
Access-Control-Max-Age: 3600
Access-Control-Allow-Headers: x-requested-with, content-type, X-AUTH-TOKEN
Access-Control-Expose-Headers: X-AUTH-TOKEN
X-Content-Type-Options: nosniff
X-Xss-Protection: 1; mode=block
Cache-Control: no-cache, no-store, max-age=0, must-revalidate
Pragma: no-cache
Expires: 0
Strict-Transport-Security: max-age=31536000 ; includeSubDomains
X-Frame-Options: DENY
Content-Type: application/json;charset=UTF-8
Transfer-Encoding: chunked
Date: Sun, 01 May 2016 12:30:03 GMT
Via: 1.1 vegur

{
 "timestamp":1462105803298,
 "status":401,
 "error":"Unauthorized",
 "message":"Authentication Failed: Bad credentials",
 "path":"/api/login"
}
```

5 Projektorganisation

A Anhang

A.1 API Aufruf: Registrieren eines neuen Benutzers

Bezeichnung	Registrieren eines neuen Benutzers
URL	`/api/register`
Methode	POST

Parameter	`{`
	`"username":[String],`
	`"password":[alphanumeric]`
	`}`

Erfolgsantwort	`HTTP/1.1 200 OK`
	`X-Content-Type-Options: nosniff`
	`X-XSS-Protection: 1; mode=block`
	`Cache-Control: no-cache, no-store, max-age=0, must-revalidate`
	`Pragma: no-cache`
	`Expires: 0`
	`X-Frame-Options: DENY`
	`X-AUTH-TOKEN: eyJpZCI6MTIsInVz...`
	`Content-Type: application/json;charset=UTF-8`
	`Content-Length: 72`
	
	`user successfully registered`

Fehlerantwort	`HTTP/1.1 422 Unprocessable Entity`
	`X-Content-Type-Options: nosniff`
	`X-XSS-Protection: 1; mode=block`
	`Cache-Control: no-cache, no-store, max-age=0, must-revalidate`
	`Pragma: no-cache`
	`Expires: 0`
	`X-Frame-Options: DENY`
	`Content-Type: text/plain;charset=UTF-8`
	`Content-Length: 17`
	
	`password to short`

Beispiel	`$ curl 'http://localhost:8080/api/register' -i -X POST -H 'Content-Type: application/`
	`json' -d`
	`'{"username":"testuser", "password": "testpassword"}'`

A.2 API Aufruf: Als registrierter Benutzer anmelden

Bezeichnung	Als registrierter Benutzer anmelden
URL	`/api/login`
Methode	POST
Parameter	`{ "username":[String], "password":[alphanumeric] }`

Erfolgsantwort	`HTTP/1.1 200 OK`
	`X-Content-Type-Options: nosniff`
	`X-XSS-Protection: 1; mode=block`
	`Cache-Control: no-cache, no-store, max-age=0, must-revalidate`
	`Pragma: no-cache`
	`Expires: 0`
	`X-Frame-Options: DENY`
	`X-AUTH-TOKEN: eyJpZCI6MTEsInVz...`

Fehlerantwort	`HTTP/1.1 401 Unauthorized`
	`X-Content-Type-Options: nosniff`
	`X-XSS-Protection: 1; mode=block`
	`Cache-Control: no-cache, no-store, max-age=0, must-revalidate`
	`Pragma: no-cache`
	`Expires: 0`
	`X-Frame-Options: DENY`

Beispiel	`$ curl 'http://localhost:8080/api/login' -i -X POST -H 'Content-Type: application/ json' -d '{"username":"user", "password": "user"}'`

A.3 API Aufruf: Einen Raum registrieren

Bezeichnung	Einen Raum registrieren
URL	`/api/rooms/:id`
Methode	PUT
URL-Parameter	**erforderlich:** `id=[String]` Beispiel: id=d934eb20-4c6f-4d1c-91c5-61cdeddcf843

Parameter	`{`
	`"name":[String]`
	`}`

Erfolgsantwort	`HTTP/1.1 200 OK`
	`X-Content-Type-Options: nosniff`
	`X-XSS-Protection: 1; mode=block`
	`Cache-Control: no-cache, no-store, max-age=0, must-revalidate`
	`Pragma: no-cache`
	`Expires: 0`
	`X-Frame-Options: DENY`
	`Content-Type: text/plain;charset=UTF-8`
	`Content-Length: 28`
	
	`room successfully registered`

Beispiel	`$ curl 'http://localhost:8080/api/rooms/d934eb20-4c6f-4d1c-91c5-61cdeddcf843' -i -X PUT -H 'Content-Type: application/json' -H 'X-AUTH-TOKEN: eyJpZCI6MTEsInVzZXJuYW...' -d '{"name":"Wohnzimmer"}'`

A.4 API Aufruf: Einen Raum von der Datenbank entfernen

Bezeichnung	Einen spezifischen Raum von der Datenbank entfernen
URL	`/api/rooms/:id`
Methode	DELETE
URL-Parameter	**erforderlich:** `id=[alphanumeric]` Beispiel: id=d934eb20-4c6f-4d1c-91c5-61cdeddcf843

Erfolgsantwort	`HTTP/1.1 200 OK`
	`X-Content-Type-Options: nosniff`
	`X-XSS-Protection: 1; mode=block`
	`Cache-Control: no-cache, no-store, max-age=0, must-revalidate`
	`Pragma: no-cache`
	`Expires: 0`
	`X-Frame-Options: DENY`
	`Content-Type: text/plain;charset=UTF-8`
	`Content-Length: 25`
	
	`room successfully removed`

Fehlerantwort	`HTTP/1.1 404 Not Found`
	`X-Content-Type-Options: nosniff`
	`X-XSS-Protection: 1; mode=block`
	`Cache-Control: no-cache, no-store, max-age=0, must-revalidate`
	`Pragma: no-cache`
	`Expires: 0`
	`X-Frame-Options: DENY`
	`Content-Type: text/plain;charset=UTF-8`
	`Content-Length: 14`
	
	`room not found`

Beispiel	`$ curl 'http://localhost:8080/api/rooms/d934eb20-4c6f-4d1c-91c5-61cdeddcf843' -i -X DELETE -H 'X-AUTH-TOKEN: eyJpZCI6MTEsInVzZXJuYW1l...'`

A.5 API Aufruf: Einen spezifischen Raum abrufen

Bezeichnung	Einen spezifischen Raum abrufen
URL	`/api/rooms/:id`
Methode	GET
URL-Parameter	**erforderlich:** `id=[alphanumeric]` Beispiel: id=d934eb20-4c6f-4d1c-91c5-61cdeddcf843

Erfolgsantwort	`HTTP/1.1 200 OK`
	`X-Content-Type-Options: nosniff`
	`X-XSS-Protection: 1; mode=block`
	`Cache-Control: no-cache, no-store, max-age=0, must-revalidate`
	`Pragma: no-cache`
	`Expires: 0`
	`X-Frame-Options: DENY`
	`Content-Type: application/json;charset=UTF-8`
	`Content-Length: 69`
	
	`"roomId":"d934eb20-4c6f-4d1c-91c5-61cdeddcf843","name":"Wohnzimmer"`

Fehlerantwort	`HTTP/1.1 404 Not Found`
	`X-Content-Type-Options: nosniff`
	`X-XSS-Protection: 1; mode=block`
	`Cache-Control: no-cache, no-store, max-age=0, must-revalidate`
	`Pragma: no-cache`
	`Expires: 0`
	`X-Frame-Options: DENY`

Beispiel	`$ curl 'http://localhost:8080/api/rooms/d934eb20-4c6f-4d1c-91c5-61cdeddcf843' -i -H 'X-AUTH-TOKEN: eyJpZCI6MTEsInVzZXJu...'`

A.6 API Aufruf: Eine Liste aller Räume abrufen

Bezeichnung	Eine Liste aller Räume abrufen
URL	`/api/rooms`
Methode	GET

Erfolgsantwort	<pre>HTTP/1.1 200 OK
	X-Content-Type-Options: nosniff
	X-XSS-Protection: 1; mode=block
	Cache-Control: no-cache, no-store, max-age=0, must-revalidate
	Pragma: no-cache
	Expires: 0
	X-Frame-Options: DENY
	Content-Type: application/json;charset=UTF-8
	Content-Length: 71
	
	[{"roomId":"d934eb20-4c6f-4d1c-91c5-61cdeddcf843","name":"Wohnzimmer"}]</pre>

Beispiel	<pre>$ curl 'http://localhost:8080/api/rooms' -i -H 'X-AUTH-TOKEN: eyJpZCI6MTEsInVzZ67KDv
	...'</pre>

A.7 API Aufruf: Ein neues Gerät registrieren

Bezeichnung	Ein neues Gerät registrieren
URL	`/api/rooms/:roomId/devices/:deviceId`
Methode	PUT

URL-Parameter	**erforderlich:**
	`roomId=[alphanumeric]`
	Beispiel: id=d934eb20-4c6f-4d1c-91c5-61cdeddcf843
	
	`deviceId=[alphanumeric]`
	Beispiel: id=a934eb20-4c6f-4d1c-91c5-61cdeddcf843

Parameter	`{`
	`"type":["HEATING" | "LIGHT"],`
	`"name":[String]`
	`}`

Erfolgsantwort	`HTTP/1.1 200 OK`
	`X-Content-Type-Options: nosniff`
	`X-XSS-Protection: 1; mode=block`
	`Cache-Control: no-cache, no-store, max-age=0, must-revalidate`
	`Expires: 0`
	`X-Frame-Options: DENY`
	`Content-Type: text/plain;charset=UTF-8`
	`Content-Length: 17`
	
	`device registered`

Fehlerantwort	`HTTP/1.1 404 Not Found`
	`X-Content-Type-Options: nosniff`
	`X-XSS-Protection: 1; mode=block`
	`Cache-Control: no-cache, no-store, max-age=0, must-revalidate`
	`Expires: 0`
	`X-Frame-Options: DENY`
	`Content-Type: text/plain;charset=UTF-8`
	`Content-Length: 14`
	
	`room not found`

Beispiel	`$ curl 'http://localhost:8080/api/rooms/d934eb20-4c6f-4d1c-91c5-61cdeddcf843/devices/a614eb20-4c6f-4d1c-91c5-61cdeddcf843' -i -X PUT -H 'Content-Type: application/json' -H 'X-AUTH-TOKEN: eyJpZCI6MT...' -d '{"deviceId":"a614eb20-4c6f-4d1c-91c5-61cdeddcf843","name":"Licht","type":"LIGHT"}'`

A.8 API Aufruf: Ein spezifisches Gerät abrufen

Bezeichnung	Ein spezifisches Gerät abrufen
URL	`/api/rooms/:roomId/devices/:deviceId`
Methode	GET

URL-Parameter	**erforderlich:**
	`roomId=[alphanumeric]`
	Beispiel: id=d934eb20-4c6f-4d1c-91c5-61cdeddcf843
	
	`deviceId=[alphanumeric]`
	Beispiel: id=a934eb20-4c6f-4d1c-91c5-61cdeddcf843

Erfolgsantwort	`HTTP/1.1 200 OK`
	`X-Content-Type-Options: nosniff`
	`X-XSS-Protection: 1; mode=block`
	`Cache-Control: no-cache, no-store, max-age=0, must-revalidate`
	`Pragma: no-cache`
	`Expires: 0`
	`X-Frame-Options: DENY`
	`Content-Type: application/json;charset=UTF-8`
	`Content-Length: 81`
	
	`{"deviceId":"a614eb20-4c6f-4d1c-91c5-61cdeddcf843","name":"Licht","type":"LIGHT"}`

Fehlerantwort	`HTTP/1.1 404 Not Found`
	`X-Content-Type-Options: nosniff`
	`X-XSS-Protection: 1; mode=block`
	`Cache-Control: no-cache, no-store, max-age=0, must-revalidate`
	`Pragma: no-cache`
	`Expires: 0`
	`X-Frame-Options: DENY`

Beispiel	`$ curl 'http://localhost:8080/api/rooms/d934eb20-4c6f-4d1c-91c5-61cdeddcf843/devices /a614eb20-4c6f-4d1c-91c5-61cdeddcf843' -i -H 'X-AUTH-TOKEN: eyJpZCI6MTEsInVzZXJuYW1lIjoi...'`

A.9 API Aufruf: Ein Gerät aus der Datenbank entfernen

Bezeichnung	Ein spezifisches Gerät aus der Datenbank entfernen
URL	`/api/rooms/:roomId/devices/:deviceId`
Methode	DELETE

URL-Parameter	**erforderlich:**
	`roomId=[alphanumeric]`
	Beispiel: id=d934eb20-4c6f-4d1c-91c5-61cdeddcf843
	
	`deviceId=[alphanumeric]`
	Beispiel: id=a934eb20-4c6f-4d1c-91c5-61cdeddcf843

Erfolgsantwort	`HTTP/1.1 200 OK`
	`X-Content-Type-Options: nosniff`
	`X-XSS-Protection: 1; mode=block`
	`Cache-Control: no-cache, no-store, max-age=0, must-revalidate`
	`Pragma: no-cache`
	`Expires: 0`
	`X-Frame-Options: DENY`
	`Content-Type: text/plain;charset=UTF-8`
	`Content-Length: 27`
	
	`device successfully removed`

Fehlerantwort	`HTTP/1.1 404 Not Found`
	`X-Content-Type-Options: nosniff`
	`X-XSS-Protection: 1; mode=block`
	`Cache-Control: no-cache, no-store, max-age=0, must-revalidate`
	`Pragma: no-cache`
	`Expires: 0`
	`X-Frame-Options: DENY`
	`Content-Type: text/plain;charset=UTF-8`
	`Content-Length: 16`
	
	`device not found`

Beispiel	`$ curl 'http://localhost:8080/api/rooms/d934eb20-4c6f-4d1c-91c5-61cdeddcf843/devices /a614eb20-4c6f-4d1c-91c5-61cdeddcf843' -i -X DELETE -H 'X-AUTH-TOKEN: eyJpZCI6MTEsInVzZXJuYW1...'`

A.10 API Aufruf: Alle Geräte eines Raumes auflisten

Bezeichnung	Alle Geräte eines Raumes auflisten
URL	`/api/rooms/:id/devices`
Methode	GET
URL-Parameter	**erforderlich:** `id=[alphanumeric]` Beispiel: id=d934eb20-4c6f-4d1c-91c5-61cdeddcf843

Erfolgsantwort	`HTTP/1.1 200 OK`
	`X-Content-Type-Options: nosniff`
	`X-XSS-Protection: 1; mode=block`
	`Cache-Control: no-cache, no-store, max-age=0, must-revalidate`
	`Pragma: no-cache`
	`Expires: 0`
	`X-Frame-Options: DENY`
	`Content-Type: application/json;charset=UTF-8`
	`Content-Length: 83`
	
	`[{"deviceId":"a614eb20-4c6f-4d1c-91c5-61cdeddcf843","name":"Licht","type":"LIGHT"}]`

Fehlerantwort	`HTTP/1.1 404 Not Found`
	`X-Content-Type-Options: nosniff`
	`X-XSS-Protection: 1; mode=block`
	`Cache-Control: no-cache, no-store, max-age=0, must-revalidate`
	`Pragma: no-cache`
	`Expires: 0`
	`X-Frame-Options: DENY`

Beispiel	`$ curl 'http://localhost:8080/api/rooms/d934eb20-4c6f-4d1c-91c5-61cdeddcf843/devices' -i -H 'X-AUTH-TOKEN: eyJpZCI6MTEsIyu2ZOL67KDvf1RPcmfC8dA...'`

A.11 API Aufruf: Verändern der Gerätedaten

Bezeichnung	Verändern von Gerätedaten
URL	`/api/rooms/:roomId/devices/:deviceId`
Methode	PATCH

URL-Parameter	**erforderlich:**
	`roomId=[alphanumeric]`
	Beispiel: id=d934eb20-4c6f-4d1c-91c5-61cdeddcf843
	
	`deviceId=[alphanumeric]`
	Beispiel: id=a934eb20-4c6f-4d1c-91c5-61cdeddcf843

Parameter	`{`
	`"targetValue":[numeric],`
	`"value":[numeric]`
	`}`

Erfolgsantwort	`HTTP/1.1 200 OK`
	`X-Content-Type-Options: nosniff`
	`X-XSS-Protection: 1; mode=block`
	`Cache-Control: no-cache, no-store, max-age=0, must-revalidate`
	`Expires: 0`
	`X-Frame-Options: DENY`
	`Content-Type: text/plain;charset=UTF-8`
	`Content-Length: 17`
	
	`device updated`

Fehlerantwort	`HTTP/1.1 404 Not Found`
	`X-Content-Type-Options: nosniff`
	`X-XSS-Protection: 1; mode=block`
	`Cache-Control: no-cache, no-store, max-age=0, must-revalidate`
	`Expires: 0`
	`X-Frame-Options: DENY`
	`Content-Type: text/plain;charset=UTF-8`
	`Content-Length: 14`
	
	`room not found`

Beispiel	`$ curl 'http://localhost:8080/api/rooms/d934eb20-4c6f-4d1c-91c5-61cdeddcf843/devices/a614eb20-4c6f-4d1c-91c5-61cdeddcf843' -i -X PATCH -H 'Content-Type: application/json' -H 'X-AUTH-TOKEN: eyJpZCI6MT...' -d '{"deviceId":"a614eb20-4c6f-4d1c-91c5-61cdeddcf843","targetValue":"0"}'`