HHN
HOCHSCHULE HEILBRONN
TECHNIK WIRTSCHAFT INFORMATIK

Software Engineering Bachelor

Embedded Systems

Projektplan

Heizungssteuerung mit ZigBee

Eugen Schlecht, Florens Hückstädt, Florian Wohlgemuth, Patrick Wohlgemuth

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Listings

Abkürzungsverzeichnis

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 Termine

5 Projektorganisation

6 Backend

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

6.1 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.

6.1.1 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.

6.1.2 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’
```

6.1.3 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: eyJpZCI6MTEsInVzZXJuYW1lIjoidXNlciIsImV4cGlyZXMiOjE0NjI5Njk0NzIwMTgsInJvbGVzIjpbIlVTRVIiXX0=.
     LDd74G0yJAuQChYR9P0A0mThfylOGGlY19u2DcTcXyQ=
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.

6.1.4 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"
}
```

A Anhang

A.1 API Call: Register as a new user

Title	Register as a new user After a successful registration the user is authenticated. The response contains a X-AUTH-TOKEN-Header for future api requests.
URL	`/api/register`
Method	POST

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

Success Response	`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`

Error Response	`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`

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

A.2 API Call: Login

Title	Login as existing user The login request must contain username and password credentials. The response contains a X-AUTH-TOKEN-Header for future api requests.
URL	`/api/login`
Method	POST

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

Success Response	`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...`

Error Response	`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`

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

A.3 API Call: Register a new room

Title	**Register a new room** The user musst send a valid X-AUTH-TOKEN in order to register a new room. The room will be linked to the user.
URL	`/api/rooms/:id`
Method	PUT
URL Params	**Required:** `id=[String]` example: id=d934eb20-4c6f-4d1c-91c5-61cdeddcf843

Data Params	`{`
	`"name":[String]`
	`}`

Success Response	`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`

Sample Call	`$ 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 Call: Delete a room

Title	**Delete a room** The user must send a valid X-AUTH-TOKEN to be able to delete a room. All registered devices will be deleted, too.
URL	`/api/rooms/:id`
Method	DELETE
URL Params	**Required:** `id=[alphanumeric]` example: id=d934eb20-4c6f-4d1c-91c5-61cdeddcf843

Success Response	`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`

Error Response	`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`

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

A.5 API Call: Show a specific room

Title	**Get a specific room object** The user must send a valid X-AUTH-TOKEN to be able to see a specific room.
URL	`/api/rooms/:id`
Method	GET
URL Params	**Required:** `id=[alphanumeric]` example: id=d934eb20-4c6f-4d1c-91c5-61cdeddcf843

Success Response	`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"`

Error Response	`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`

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

A.6 API Call: Show all rooms

<table>
<tr><td><b>Title</b></td><td><b>Show all rooms</b>
The user must send a valid X-AUTH-TOKEN to be able to see his rooms.</td></tr>
<tr><td><b>URL</b></td><td><code>/api/rooms</code></td></tr>
<tr><td>Method</td><td>GET</td></tr>
<tr><td>Success Response</td><td>

```
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"}]
```

</td></tr>
<tr><td>Sample Call</td><td>

```
$ curl ’http://localhost:8080/api/rooms’ -i -H ’X-AUTH-TOKEN:
    eyJpZCI6MTEsInVzZ67KDv...’
```

</td></tr>
</table>

A.7 API Call: Register a new device

Title	**Add a new Device to a Room**
URL	`/api/rooms/:roomId/devices/:deviceId`
Method	PUT

URL Params	**Required:**
	`roomId=[alphanumeric]`
	example: id=d934eb20-4c6f-4d1c-91c5-61cdeddcf843
	
	`deviceId=[alphanumeric]`
	example: id=a934eb20-4c6f-4d1c-91c5-61cdeddcf843

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

Success Response	`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`

Error Response	`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`

Sample Call	`$ 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 Call: Show a specific device

Title	**Show a specific device of a room**
URL	`/api/rooms/:roomId/devices/:deviceId`
Method	GET

URL Params	**Required:**
	`roomId=[alphanumeric]`
	example: id=d934eb20-4c6f-4d1c-91c5-61cdeddcf843
	
	`deviceId=[alphanumeric]`
	example: id=a934eb20-4c6f-4d1c-91c5-61cdeddcf843

Success Response	`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"}`

Error Response	`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`

Sample Call	`$ 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 Call: Delete a specific device

Title	**Delete a device**
URL	`/api/rooms/:roomId/devices/:deviceId`
Method	DELETE

URL Params	**Required:**
	`roomId=[alphanumeric]`
	example: id=d934eb20-4c6f-4d1c-91c5-61cdeddcf843
	`deviceId=[alphanumeric]`
	example: id=a934eb20-4c6f-4d1c-91c5-61cdeddcf843

Success Response	`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`

Error Response	`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`

Sample Call	`$ 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 Call: Show all devices

Title	**Show all devices of a room**
URL	`/api/rooms/:id/devices`
Method	GET
URL Params	**Required:** `id=[alphanumeric]` example: id=d934eb20-4c6f-4d1c-91c5-61cdeddcf843

Success Response	`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`
	`    "}]`

Error Response	`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`

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

A.11 API Call: Send a command to a specific device

Title	**Send a command to a specific device**
URL	`/api/rooms/:roomId/devices/:deviceId/command`
Method	POST

Data Params	`{`
	`"type":["SET" | "GET"],`
	`"value":[numeric],`
	`}`

Success Response	`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`
	
	`command successfully sent`

Error Response	`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`

Sample Call	`$ curl ’http://localhost:8080/api/rooms/d934eb20-4c6f-4d1c-91c5-61cdeddcf843/`
	`    devices/a614eb20-4c6f-4d1c-91c5-61cdeddcf843/command’ -i -X POST -H ’Content-`
	`    Type: application/json’ -H ’X-AUTH-TOKEN:`
	`    eyJpZCI6MTEsInVzZXJuYW1lIjoidXNlciIsImV4...’ -d ’{"type":"SET","value":20.0}’`