HOCHSCHULE FULDA
UNIVERSITY OF APPLIED SCIENCES

WEB-APPLIKATIONEN

ROUTING

AI
Manuel Richardt, Dominik Rupprecht
FB **ANGEWANDTE INFORMATIK**

ROUTING IN NODE.JS

ROUTEN WERDEN IN DER APP.JS EINGETRAGEN

Manuel Richardt, Dominik Rupprecht
AI FB ANGEWANDTE INFORMATIK

ROUTING

› Routing bezieht sich auf Anwendungsendpunkte (URIs) und deren Antworten auf Clientanforderungen

```
router.get('/', function(req, res, next) {
  res.send('respond with a resource');
});
```

› „get“ ist eine http-Methode (Erläuterung auf kommenden Folien)

› „/“ Definiert im oberen Bild die Route. Im Browser oder ThunderClient erreichen Sie die Route unter http://localhost:3000/

PARAMETER IN ROUTEN
LIVE-CODE

› Parameter in der URI

http://localhost/altersberechnung?birthdate=2022-03-14

```
router.get('/altersberechnung', function(req, res, next) {
  let bdate = req.query.birthdate;
```

› Parameter als Route

http://localhost/users/Manuel

```
router.get('/:name', function(req, res, next) {
  console.log(req.params.name);
  res.send('Hallo '+req.params.name);
});
```

`:name` bindet einen Teil der URI („Manuel“) an den Parameter `name`

AI
Manuel Richardt, Dominik Rupprecht
FB **ANGEWANDTE INFORMATIK**

PARAMETER AUSLESEN
LIVE-CODE UND THUNDERCLIENT

› GET → Query

› POST → Body

› URI → Params

```
router.get('/', function(req, res, next) {
  console.log(req.query.name)
  res.send('respond with a resource');
});
router.post('/', function(req, res, next) {
  console.log(req.body.name);
  res.send('respond post with a resource');
});
router.get('/:name', function(req, res, next) {
  console.log(req.params.name);
  res.send('Hallo '+req.params.name);
});
```

HOCHSCHULE FULDA
UNIVERSITY OF APPLIED SCIENCES

WEB-PROGRAMMIERUNG

HTTP-METHODEN

AI
Manuel Richardt, Dominik Rupprecht
FB **ANGEWANDTE INFORMATIK**

HTTP-METHODEN

Safe

› Der Aufruf dieser Operation hat keine Nebenwirkungen.

› Ressource erfährt durch den Abruf keine Änderung

Idempotent

› Das mehrfache Aufrufen dieser Operation liefert immer das gleiche Ergebnis.

HTTP METHODEN

› GET
  Anforderung einer Ressource
  Webservice liefert die Repräsentation einer Ressource
  Ist safe und idempotent

› POST
  Erzeugt eine oder mehrere Ressourcen deren URIs noch nicht bekannt sind.
  Bietet die meiste Flexibilität, da sie weder safe noch idempotent ist

› Daneben gibt es noch:
  HEAD, PUT, DELETE, CONNECT, OPTIONS, TRACE und PATCH

ZUSAMMENFASSUNG

› Routendateien werden zunächst deklariert und dann unter einem Aufrufpfad eingebunden.

› Routen werden (i.d.R.) für explizite HTTP Methoden geschrieben.

› Parameter können unterschiedlich übertragen und ausgelesen werden