PRÄSENTIERT

Sicherheit für REST-Webservice-Schnittstellen implementieren

Dipl. Wirt-Ing.
Dirk Neumann
Neumann Softwaretechnik Consulting / Bochum
http://www.Neumann-Software.de

EIN BEITRAG VON

Dipl. Wirt-Ing.

Dirk Neumann

Technology Mentor:

- Ermittlung von Modernisierungsmöglichkeiten
- Individuelle Bewertung von Vorgehensweisen, Methoden und Werkzeugen bei der Softwareentwicklung
- Zusammenstellung von Lösungskomponenten für Ihre Anforderungen
- Einführung neuer Methoden und Werkzeuge
- Coaching des Projekt- und Entwicklerteams
- Qualitätsmanagement der Entwicklung und des Betriebs

Neumann Softwaretechnik Consulting / Bochum

http://www.Neumann-Software.de

POW3R
einfach. digital. agil.

Sicherheit für REST-Webservice-Schnittstellen implementieren

Abstract:

Datenaustausch mit Webservices ist in aller Munde. Besonders beliebt sind REST-Webservices.

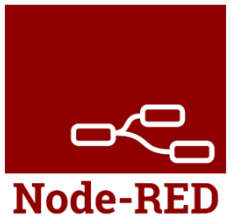
„Das ist schnell gemacht.“

Aber bedeutet „schnell“ auch „sicher“, wenn man die Schnittstellen für Kunden und Lieferanten im Internet zur Verfügung stellen muss?

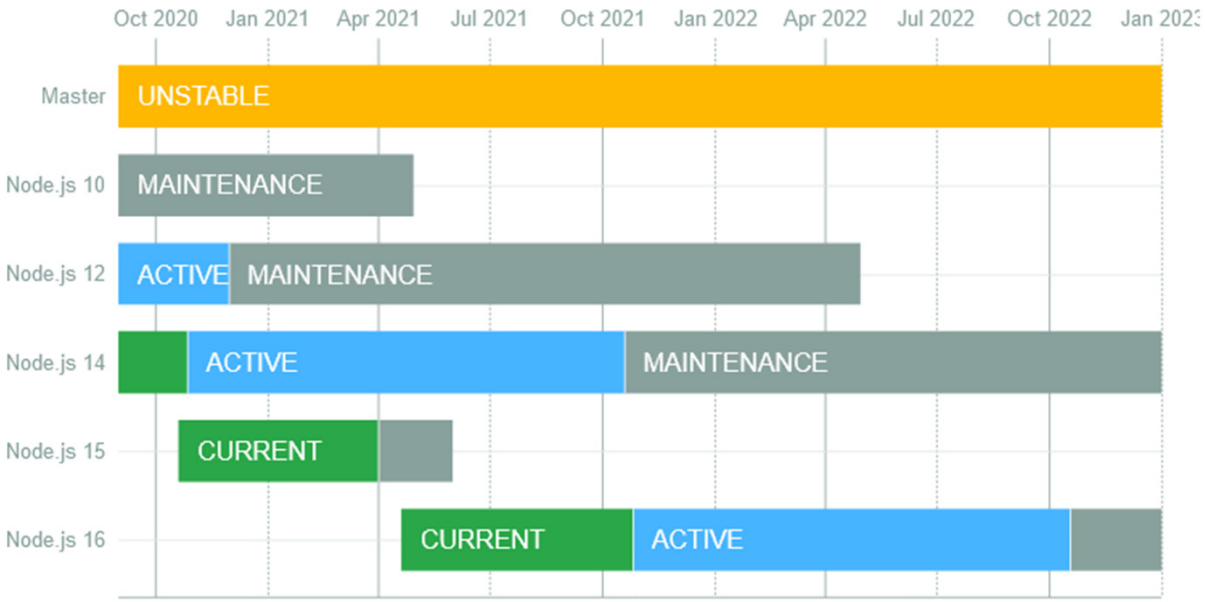
Der Vortrag zeigt anhand von konkreten node.js-Beispielen, wie ein sicherer Datenaustausch mit REST-Webservices gestaltet werden kann.

POW3R
einfach. digital. agil.

Systemumgebung



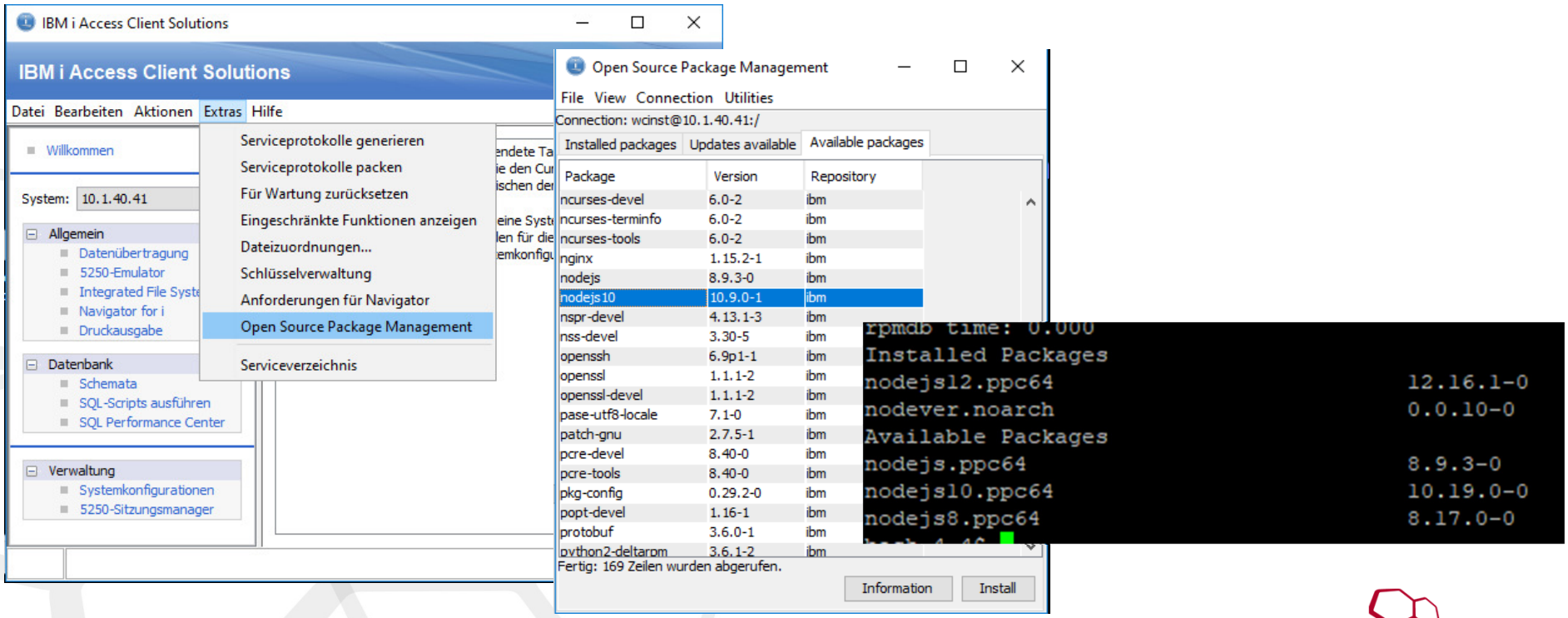
nodeJS - Releases



Quelle: https://nodejs.org/en/about/releases/

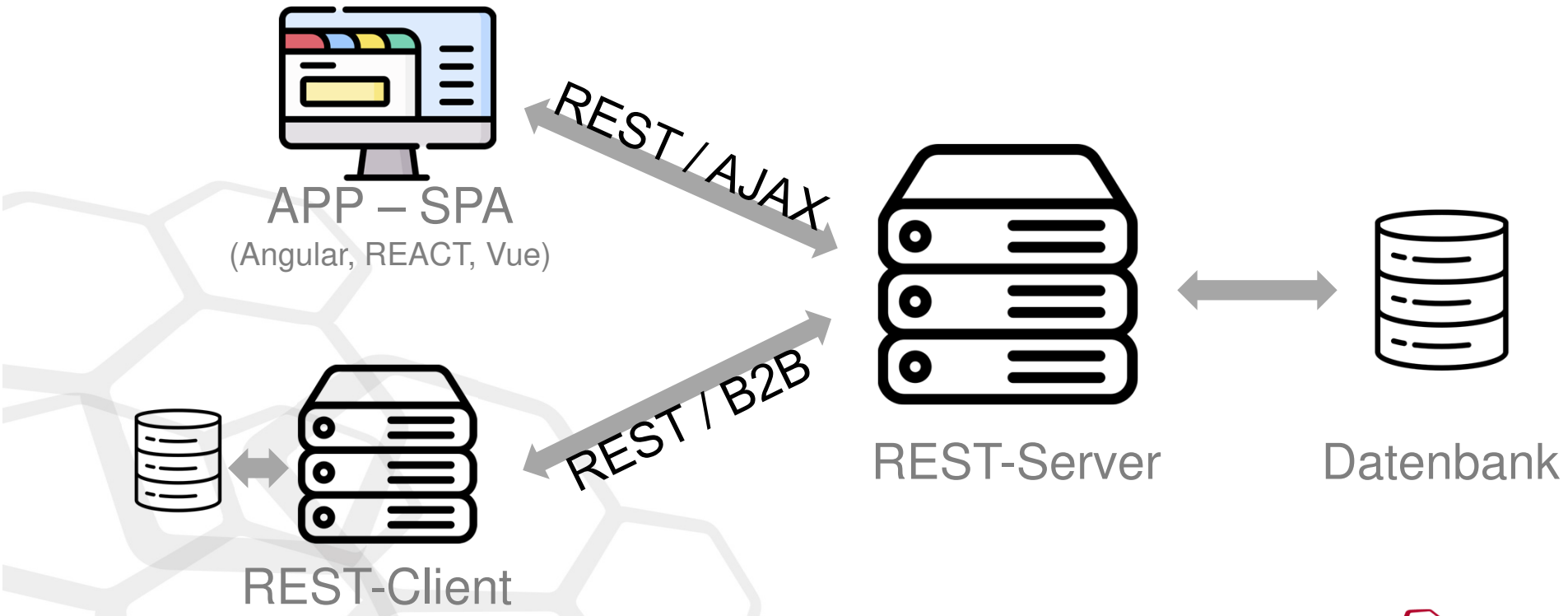
einfach. digital. agil.

IBM i - Installation Open-Source Pakte



POWER
einfach. digital. agil.

Verwendung von REST-Webservices



Icons made by Freepik, Pixel perfect, catkuro from www.flaticon.com

POW3R
einfach. digital. agil.

REST Architektur

- REST steht für Representational State Transfer
  definiert in der Dissertation von Roy Fielding in 2000

- 6 Architektur Anforderungen:
  - Uniform interface
  - Client-server
  - Stateless
  - Cacheable
  - Layered system
  - Code on demand (optional)

Fielding, Roy Thomas (2000). "Architectural Styles and the Design of Network-based Software Architectures". Dissertation. University of California, Irvine.

POW3R
einfach. digital. agil.

REST-Befehle

Befehl (HTTP-Methode)	Beschreibung
CONNECT	Dient dazu, die Anfrage durch einen TCP-Tunnel zu leiten. Wird meist eingesetzt, um eine HTTPS-Verbindung über einen HTTP-Proxy herzustellen.
DELETE	löscht die angegebene Ressource.
GET	fordert die angegebene Ressource vom Server an. GET weist keine Nebeneffekte auf. Der Zustand am Server wird nicht verändert, weshalb GET als sicher bezeichnet wird.
HEAD	fordert Metadaten zu einer Ressource an.
OPTIONS	prüft, welche Methoden auf einer Ressource zur Verfügung stehen.
PATCH	ein Teil der angegebenen Ressource wird geändert. Hierbei sind Nebeneffekte erlaubt.
POST	fügt eine neue (Sub-)Ressource unterhalb der angegebenen Ressource ein. Da die neue Ressource noch keinen URI besitzt, adressiert der URI die übergeordnete Ressource. Als Ergebnis wird der neue Ressourcenlink dem Client zurückgegeben. POST kann im weiteren Sinne auch dazu verwendet werden, Operationen abzubilden, die von keiner anderen Methode abgedeckt werden.
PUT	die angegebene Ressource wird angelegt. Wenn die Ressource bereits existiert, wird sie geändert.
TRACE	Gibt die Anfrage zurück, wie sie der Zielserver erhält. Dient etwa dazu, um Änderungen der Anfrage durch Proxyserver zu ermitteln.

POW3R
einfach. digital. agil.

REST Security Designprinzipien

Economy of mechanism: Keep the design as simple and small as possible

Fail-safe defaults: Base access decisions on permission rather than exclusion

Complete mediation: Every access to every object must be checked for authority

Open design: The design should not be secret

Separation of privilege: Where feasible, a protection mechanism that requires two keys to unlock it is more robust and flexible than one that allows access to the presenter of only a single key.

Least privilege: Every program and every user of the system should operate using the least set of privileges necessary to complete the job

Least common mechanism: Minimize the amount of mechanism common to more than one user and depended on by all users

Psychological acceptability: It is essential that the human interface be designed for ease of use, so that users routinely and automatically apply the protection mechanisms correctly

Quelle:
The Protection of Information in Computer Systems
JEROME H. SALTZER, SENIOR MEMBER, IEEE, AND
MICHAEL D. SCHROEDER, MEMBER, IEEE
http://web.mit.edu/Saltzer/www/publications/protection/Basic.html

POW3R
einfach. digital. agil.

REST-URL Beispiele

HTTP GET http://rest-api.server.com/products
//Alle Produkte

HTTP POST http://rest-api.server.com/products
//neues Produkt erstellen

HTTP GET http://rest-api.server.com/products/{id}
//Das Produkt mit der „id“ zurückgeben

HTTP PATCH http://rest-api.server.com/products/{id}
//Das Produkt mit der „id“ aktualisieren

HTTP DELETE http://rest-api.server.com/products/{id}
//Das Produkt mit der „id“ löschen

POW3R
einfach. digital. agil.

Beispiel: REST Service für Produkte

Datenbank: Produkte

PRODID	NAME	PGROUP	PRICEPUR	PRICESEL	STOCKQTY
100	Toaster A	TOASTER	74.80	169.99	43
101	Toaster B	TOASTER	24.80	89.99	23
102	Toaster Z2	TOASTER	34.80	69.99	10
103	Toaster X	TOASTER	54.80	119.99	47
104	Toaster B	TOASTER	14.80	69.99	9
200	Grill 8000	GRILL	170.80	269.99	12
210	Grill 8000X	GRILL	175.80	289.99	90
220	Grill 4000	GRILL	180.80	329.99	94
225	Grill 4000S	GRILL	185.80	359.99	37
265	Grill 1000	GRILL	190.80	419.99	71
300	Trockner S	TROCKNER	200.30	499.99	2
330	Trockner M	TROCKNER	220.80	549.99	150
358	Trockner L	TROCKNER	240.00	609.99	1
356	Trockner XL	TROCKNER	270.22	639.99	38
340	Trockner XXL	TROCKNER	300.90	749.99	82

POW3R
einfach. digital. agil.

User-Gruppen für Webservices

ADMIN
All access

Webshop
Limit 50

Kundin A
limit 10, SELPRICE, STOCK only,
format XML

Kunde B
limit 10, SELPRICE only,
json/xml

Vertriebsaußendienst 1
PGROUP COFFE only

Vertriebsaußendienst 2
PGROUP GRILL only

Vertriebsinnendienst
READ only

Icons made by itim2101, Freepik from www.flaticon.com

POW3R
einfach. digital. agil.

URL-Inflation für Usergruppen

Achtung - Schlechtes Design!

HTTP GET http://rest-api.server.com/products

HTTP GET http://rest-api.server.com/shop/products
HTTP GET http://rest-api.server.com/kundeA/products
HTTP GET http://rest-api.server.com/kundeB/products
HTTP GET http://rest-api.server.com/VTBout1/products
HTTP GET http://rest-api.server.com/VTBout2/products
HTTP GET http://rest-api.server.com/VTBin/products

POW3R
einfach. digital. agil.

REST-URL Beispiele mit Filter

Gutes Design: generische Route + Filteroptionen

HTTP GET http://rest-api.server.com/products

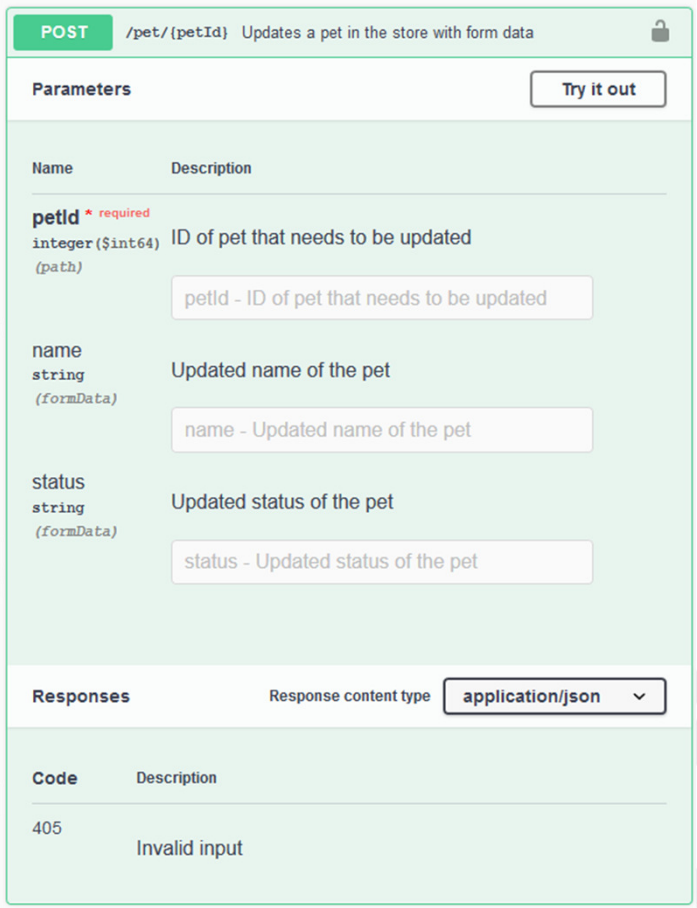
HTTP GET http://rest-api.server.com/products?offset=100&count=50

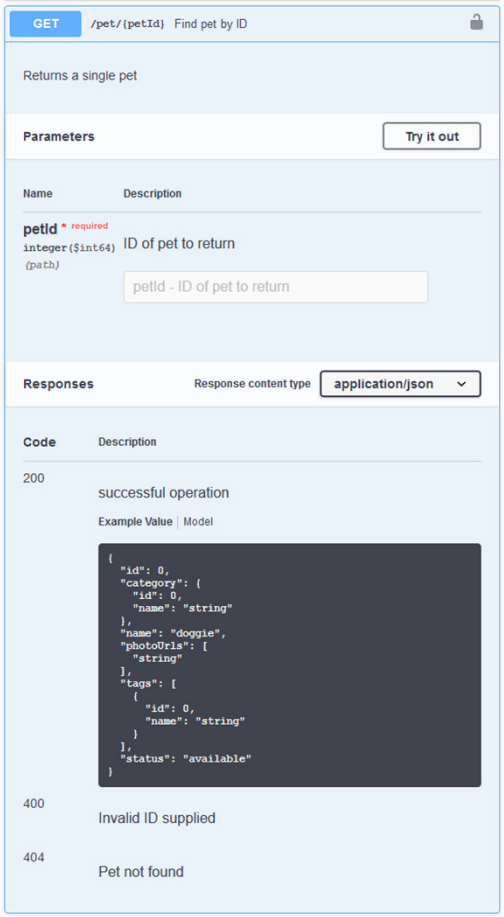
HTTP GET http://rest-api.server.com/products?pgroup=Coffee

HTTP GET http://rest-api.server.com/products?sort=price&order=desc

POW3R
einfach. digital. agil.

Swagger / OpenAPI





https://swagger.io/tools/swagger-ui/

OpenAPI Specification

OPENAPI INITIATIVE

The World Standard for RESTful APIs

https://swagger.io/resources/open-api/

POW3R
einfach. digital. agil.

Struktur einer Express-REST-Anwendung

REST

HTTP-GET
https://server.com:443/products

Express-App.js
(node.js)

Router
„GET https://…/products“

Controller
„listAction“

Model
„Products.getAll()“

Icons made by Freepik, catkuro from www.flaticon.com

POW3R
einfach. digital. agil.

Login - Route > getJWTToken

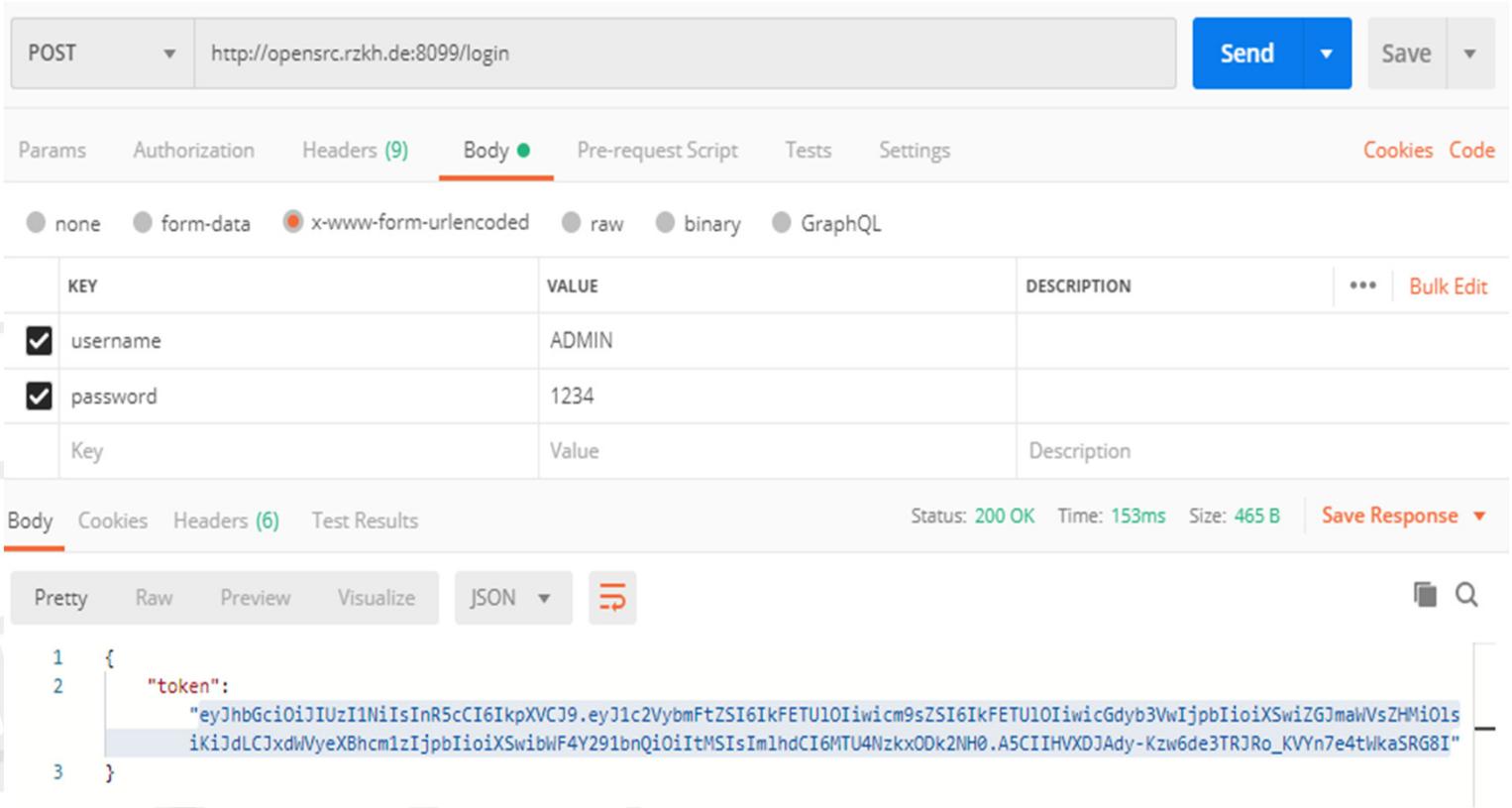
POST http://opensrc.rzkh.de:8099/login

	KEY	VALUE
☑	username	ADMIN
☑	password	1234

```
7   router.post('/', (request, response) =>{
8       userModel
9       .get({
10          username: request.body.username,
11          password: request.body.password,
12      })
13      .then(
14          user => {
15              if (user) {
16                  const payload = { ...user };
17                  delete payload.password;
18                  const token = jwt.sign(payload, 'secure');
19                  response.json({ token });
20              } else {
21                  response.status(401).json('unauthorized');
22              }
23          },
24          error => response.status(401).json('unauthorized'),
25      );
26  });
```

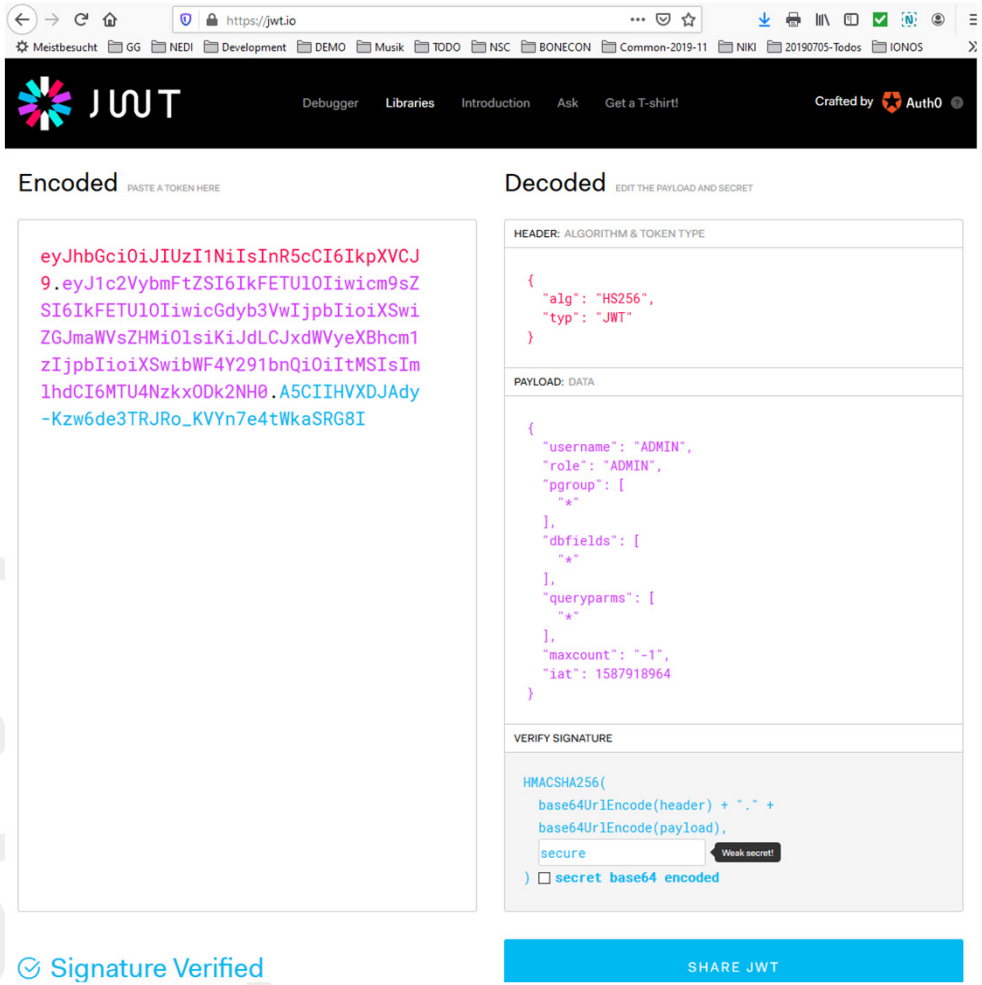
POW3R
einfach. digital. agil.

Postman: Login mit JWT-Token



POW3R
einfach. digital. agil.

Inhalt eines JWT-Tokens



POW3R
einfach. digital. agil.

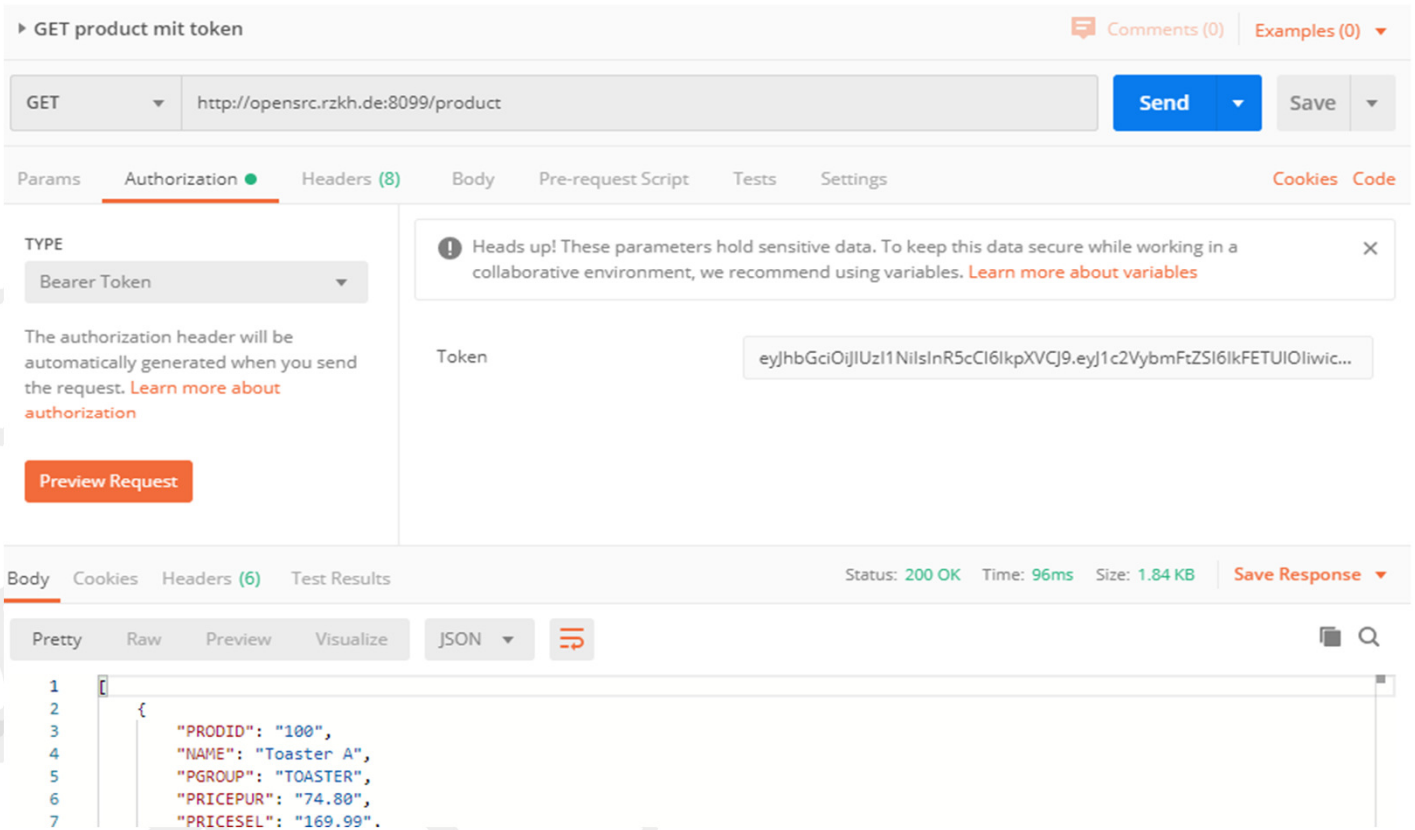
EXPRESS Middleware Module

- jwt: middleware für Prüfung isTokenValid

```
30  app.use('/login', loginRouter);
31  app.use('/products',
32      // check login/jwt-token
33      expressJWT({ secret: 'secure'}),
34      productRouter);
35
36  // Abfangen UnauthorizedError
37  app.use(function (err, request, response, next) {
38      if (err.name === 'UnauthorizedError') {
39          response.status(401).json('unauthorized');
40      } else {
41          next();
42      }
43  });
44
```

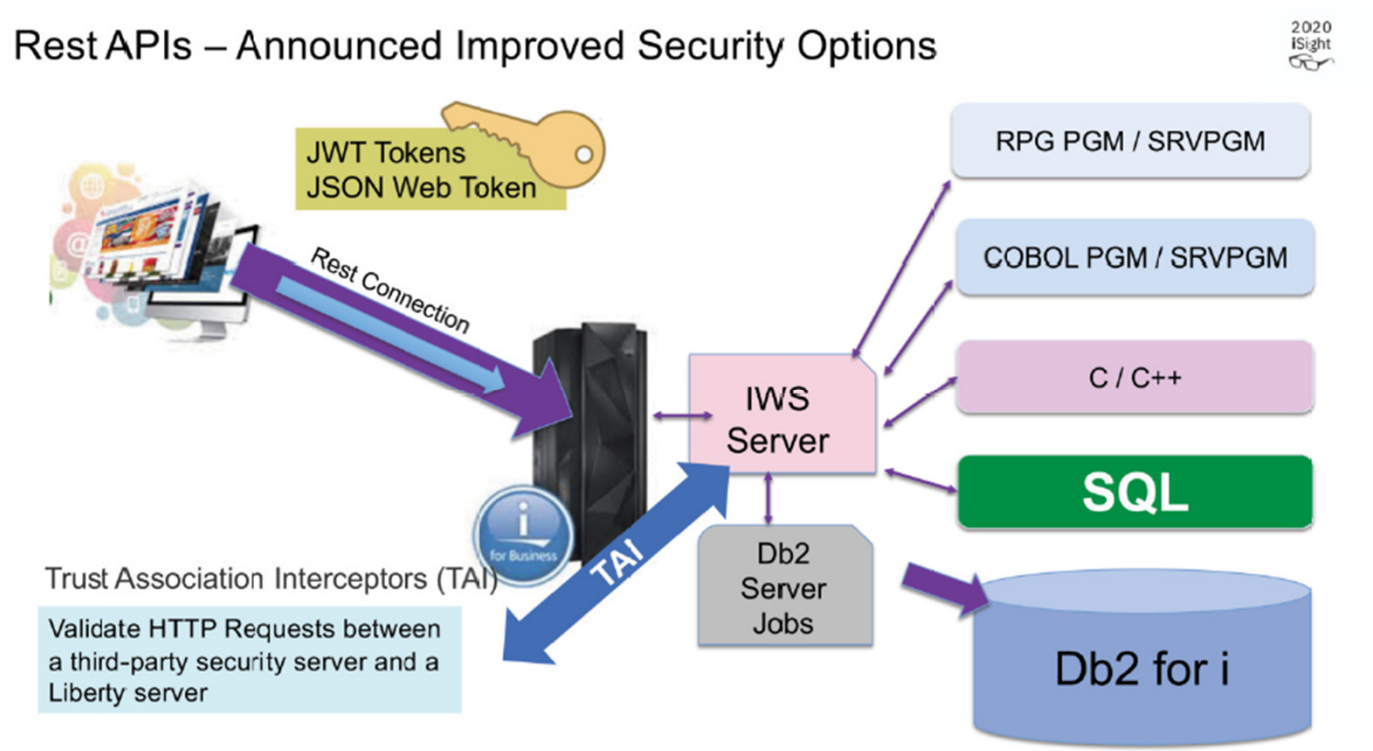
POW3R
einfach. digital. agil.

Postman: GET product mit JWT-Token



POW3R
einfach. digital. agil.

IBM i Integrated Webservice Server: JWT-Unterstützung



Quelle: IBM Announcement Technology Refresh 10/2020

POWER
einfach. digital. agil.

EXPRESS Middleware Module

- checkPriviledge
- checkQueryParms
- setMaxCount
- setFieldList
- setPGroups

```
12
13  router.get('/',
14      // Anzeigen des aktuellen User
15      middleware.displayUser,
16      middleware.checkPriviledge('productList'),
17      middleware.checkQueryParms(),
18      middleware.setMaxCount(),
19      middleware.setFieldlist(),
20      middleware.setPGroups(),
21      listAction2);
22
23  // router.get('/:prodId', detailAction);
24
```

POW3R
einfach. digital. agil.

EXPRESS Middleware Module

- checkPriviledge

```
6   function checkPriviledge(priv) {
7
8       var m_checkPriviledge = function (req, res, next) {
9           // all access
10          if (req.user.priviledges.includes('*')) {
11              // console.log('checkPriviledge: current user.priviledge = *');
12              return next()
13          };
14
15          // current priv
16          if (req.user.priviledges.includes(priv)) {
17              // console.log('checkPriviledge: current user.priviledge includes ' + priv);
18              return next();
19          } else {
20              console.log('checkPriviledge: current user.priviledge does not include ' + priv);
21              res.status(401).json('unauthorized');
22          };
23      }
24
25      return m_checkPriviledge;
26  };
```

POW3R
einfach. digital. agil.

EXPRESS Middleware Module

- checkQueryParms

```
function checkQueryParms() {

    var m_checkQueryParms = function (req, res, next) {
        const options = {
        sort : (req.query.sort
            && (req.user.queryparms.includes('*')
                || req.user.queryparms.includes('sort')))
              ? req.query.sort : '',
        order : (req.query.order
            && (req.user.queryparms.includes('*')
            || req.user.queryparms.includes('order')))
            ? req.query.order : '',
        offset : (req.query.offset
            && (req.user.queryparms.includes('*')
            || req.user.queryparms.includes('offset')))
            ? req.query.offset : '',
        count : (req.query.count
            && (req.user.queryparms.includes('*')
            || req.user.queryparms.includes('count')))
            ? req.query.count : '',
        }

        //
        // console.log('checkQueryParms: options=' + JSON.stringify(options));

        req.listAction = {};
        req.listAction.options = options;

        return next();
    }

    return m_checkQueryParms;

};
```

POW3R
einfach. digital. agil.

DB-Model

Einfach:

```
// idb-connector
const {dbconn, dbstmt} = require('idb-connector');
const connection = new dbconn();
connection.conn('*LOCAL');
const schema = 'NEDI68';

function getAll1(options) {
    return new Promise((resolve, reject) => {

        let sSql = "SELECT * FROM " + schema + ".PRODREST";

        //
        console.log(sSql);
        const statement = new dbstmt(connection);
        statement.exec(sSql, (result, err) => {
            // console.log(JSON.stringify(result));
            if (err) {reject(err)};

            resolve(result);
            statement.close();
        });

    })
}

```

POW3R
einfach. digital. agil.

Gefahr: SQL-Injection

HTTP GET http://rest-api.server.com/products?sort=price&dir=desc

tmpSql = „SELECT * FROM PRODUCTS“
tmpSql += „ORDER BY “ + parm(sort) + „ “ + parm(dir)
> SQL-String:
SELECT * FROM PRODUCTS ORDER BY PRICE DESC

HTTP GET http://rest-api.server.com/products?sort=price&dir=desc;INSERT INTO USERS VALUES (‚HACKER‘, ‚SECOFR‘, ‚PW1234‘);

SQL-String:
SELECT * FROM PRODUCTS ORDER BY PRICE DESC;INSERT INTO USERS VALUES (‚HACKER‘, ‚SECOFR‘, ‚PW1234‘);

POW3R
einfach. digital. agil.

DB-Model

erweitert:

```
26  // mit Auswertung der Queryparameter sort, order, of
27  function getAll2(options) {
28      return new Promise((resolve, reject) => {
29
30          let sSql = "SELECT "
31
32          if (options.fields.includes('*')) {
33              sSql += "*";
34          } else if (options.fields.length > 0) {
35              sSql += options.fields.join();
36          } else {
37              resolve({});
38          }
39
40          sSql += " FROM " + schema + ".PRODREST";
41
42
43          // Auswertung Product-Groups
44          if (options.pgroups.length == 0) {
45              resolve({});
46          } else if (!options.pgroups.includes('*')) {
47              sSql += " WHERE PGROUP";
48              sSql += " IN ('"
49              sSql += options.pgroups.join("', '");
50              sSql += "')"
51          }
52
```

```
54          // Auswertung Query-Parameter
55          if (options.sort) {
56              sSql += ' ORDER BY ' + options.sort;
57              if (options.order &&
58                  ['asc', 'desc'].includes(options.order.toLowerCase())) {
59                      sSql += ' ' + options.order;
60                  }
61          }
62
63          let offset = 0;
64          if (options.offset) {
65              offset = parseInt(options.offset);
66          }
67
68          let count = 0;
69          if (options.count) {
70              count = parseInt(options.count);
71          }
72
73          // check maxcount
74          let maxcount = parseInt(options.maxcount);
75          console.log(maxcount);
76          if (maxcount > -1) {
77              if (count == 0) {
78                  count = maxcount;
79              } else if (maxcount < count) {
80                  count = maxcount;
81              }
82          }
83
```

```
83          if (offset != 0 || count != 0) {
84              sSql += ' LIMIT ' + offset + ', ' + count;
85          }
86
87
88
89          //
90          console.log(sSql);
91          const statement = new dbstmt(connection);
92          statement.exec(sSql, (result, err) => {
93              // console.log(JSON.stringify(result));
94              if (err) {reject(err)};
95
96              resolve(result);
97              statement.close();
98          });
99
100     })
101 }
102
```

POW3R
einfach. digital. agil.

ADMIN

ADMIN

All access

```
const users = [
    { username: 'ADMIN',
      password: '1234',
      role: 'ADMIN',
      priviledges: ['*'],
      pgroups: ['*'],
      dbfields: ['*'],
      queryparms: ['*'],
      maxcount: '-1' } ,
```

http://opensrc.rzkh.de:8099/products?sort=STOCKQTY&order=desc

```
current user= {"username":"ADMIN","role":"ADMIN","priviledges":["*"],"pgroups":[
"*"],"dbfields":["*"],"queryparms":["*"],"maxcount":"-1","iat":1588685555}
-1
SELECT * FROM NEDI68.PRODREST ORDER BY STOCKQTY desc
```


[{"PRODID":"330","NAME":"Trockner M","PGROUP":"TROCKNER","PRICEPUR":"220.80","PRICESEL":"549.99","STOCKQTY":"150"},
{"PRODID":"220","NAME":"Grill 4000","PGROUP":"GRILL","PRICEPUR":"180.80","PRICESEL":"329.99","STOCKQTY":"94"},
{"PRODID":"210","NAME":"Grill 8000X","PGROUP":"GRILL","PRICEPUR":"175.80","PRICESEL":"289.99","STOCKQTY":"90"},
{"PRODID":"340","NAME":"Trockner XXL","PGROUP":"TROCKNER","PRICEPUR":"300.90","PRICESEL":"749.99","STOCKQTY":"82"},
{"PRODID":"265","NAME":"Grill 1000","PGROUP":"GRILL","PRICEPUR":"190.80","PRICESEL":"419.99","STOCKQTY":"71"},
{"PRODID":"103","NAME":"Toaster X","PGROUP":"TOASTER","PRICEPUR":"54.80","PRICESEL":"119.99","STOCKQTY":"47"},
{"PRODID":"100","NAME":"Toaster A","PGROUP":"TOASTER","PRICEPUR":"74.80","PRICESEL":"169.99","STOCKQTY":"43"},
{"PRODID":"356","NAME":"Trockner XL","PGROUP":"TROCKNER","PRICEPUR":"270.22","PRICESEL":"639.99","STOCKQTY":"38"},
{"PRODID":"225","NAME":"Grill 4000S","PGROUP":"GRILL","PRICEPUR":"185.80","PRICESEL":"359.99","STOCKQTY":"37"},
{"PRODID":"101","NAME":"Toaster B","PGROUP":"TOASTER","PRICEPUR":"24.80","PRICESEL":"89.99","STOCKQTY":"23"},
{"PRODID":"200","NAME":"Grill 8000","PGROUP":"GRILL","PRICEPUR":"170.80","PRICESEL":"269.99","STOCKQTY":"12"},
{"PRODID":"102","NAME":"Toaster Z2","PGROUP":"TOASTER","PRICEPUR":"34.80","PRICESEL":"69.99","STOCKQTY":"10"},
{"PRODID":"104","NAME":"Toaster B","PGROUP":"TOASTER","PRICEPUR":"14.80","PRICESEL":"69.99","STOCKQTY":"9"},
{"PRODID":"300","NAME":"Trockner S","PGROUP":"TROCKNER","PRICEPUR":"200.30","PRICESEL":"499.99","STOCKQTY":"2"},
{"PRODID":"358","NAME":"Trockner L","PGROUP":"TROCKNER","PRICEPUR":"240.00","PRICESEL":"609.99","STOCKQTY":"1"}]

POW3R

einfach. digital. agil.

AZUBI

AZUBI

no priv „productList“

```
{ username: 'AZUBI',
    password: '0000',
    role: 'AZUBI',
    priviledges: [],
    pgroups: ['*'],
    dbfields: ['*'],
    queryparms: ['*'],
    maxcount: '1' },
```

http://opensrc.rzkh.de:8099/products

```
current user= {"username":"AZUBI","role":"AZUBI","priviledges":[],"pgroups":["*"
],"dbfields":["*"],"queryparms":["*"],"maxcount":"1","iat":1588686371}
checkPriviledge: current user.priviledge does not include productList
```

Body Cookies Headers (6) Test Results Status: 401 Unauthorized

Pretty Raw Preview Visualize JSON

```
1  "unauthorized"
```

POW3R
einfach. digital. agil.

Kundin A

Kundin A

limit 10, SELPRICE, STOCK only, format XML

```
{ username: 'KDA',
    password: 'A',
    role: 'KUNDE',
    priviledges: ['productList'],
    pgroups: ['TOASTER', 'GRILL'],
    dbfields: ['PRODID',
        'PGROUP',
        'NAME',
        'PRICESEL',
        'STOCKQTY'],
    queryparms: ['count', 'offset'],
    maxcount: '5' },
```

http://opensrc.rzkh.de:8099/products
?sort=STOCKQTY&order=desc&count=20&offset=2

```
current user= {"username":"KDA","role":"KUNDE","priviledges":["productList"],"pgro
ups":["TOASTER","GRILL"],"dbfields":["PRODID","PGROUP","NAME","PRICESEL","STOCKQTY
"],"queryparms":["count","offset"],"maxcount":"5","iat":1588686832}
5
SELECT PRODID,PGROUP,NAME,PRICESEL,STOCKQTY FROM NEDI68.PRODREST WHERE PGROUP IN (
'TOASTER', 'GRILL') LIMIT 2, 5
```


[{"PRODID":"102","PGROUP":"TOASTER","NAME":"Toaster Z2","PRICESEL":"69.99","STOCKQTY":"10"},
{"PRODID":"103","PGROUP":"TOASTER","NAME":"Toaster X","PRICESEL":"119.99","STOCKQTY":"47"},
{"PRODID":"104","PGROUP":"TOASTER","NAME":"Toaster B","PRICESEL":"69.99","STOCKQTY":"9"},
{"PRODID":"200","PGROUP":"GRILL","NAME":"Grill 8000","PRICESEL":"269.99","STOCKQTY":"12"},
{"PRODID":"210","PGROUP":"GRILL","NAME":"Grill 8000X","PRICESEL":"289.99","STOCKQTY":"90"}]

POW3R
einfach. digital. agil.

Vertriebsaußendienst 1

Vertriebsaußendienst 1
PGROUP TOASTER only

```
{ username: 'VTA1',
    password: '1',
    role: 'VTB-A',
    priviledges: ['productList'],
    pgroups: ['TOASTER'],
    dbfields: ['*'],
    queryparms: ['*' ],
    maxcount: '5' },
```

http://opensrc.rzkh.de:8099/products
?sort=PRICESEL&order=desc&count=20

```
current user= {"username":"VTA1","role":"VTB-A","priviledges":["productList"],"pgr
oups":["TOASTER"],"dbfields":["*"],"queryparms":["*"],"maxcount":"5","iat":1588687
113}
5
SELECT * FROM NEDI68.PRODREST WHERE PGROUP IN ('TOASTER') ORDER BY PRICESEL desc L
IMIT 0, 5
```


[{"PRODID":"100","NAME":"Toaster A","PGROUP":"TOASTER","PRICEPUR":"74.80","PRICESEL":"169.99","STOCKQTY":"43"},
{"PRODID":"103","NAME":"Toaster X","PGROUP":"TOASTER","PRICEPUR":"54.80","PRICESEL":"119.99","STOCKQTY":"47"},
{"PRODID":"101","NAME":"Toaster B","PGROUP":"TOASTER","PRICEPUR":"24.80","PRICESEL":"89.99","STOCKQTY":"23"},
{"PRODID":"102","NAME":"Toaster Z2","PGROUP":"TOASTER","PRICEPUR":"34.80","PRICESEL":"69.99","STOCKQTY":"10"},
{"PRODID":"104","NAME":"Toaster B","PGROUP":"TOASTER","PRICEPUR":"14.80","PRICESEL":"69.99","STOCKQTY":"9"}]

POW3R
einfach. digital. agil.

Vertriebsinnendienst

Vertriebsinnendienst
READ only

```
{ username: 'VTI',
    password: 'I',
    role: 'VTB-I',
    priviledges: ['productList'],
    pgroups: ['*'],
    dbfields: ['*'],
    queryparms: ['*' ],
    maxcount: '10' },
```

http://opensrc.rzkh.de:8099/products
?sort=PRICESEL&order=desc&count=20

```
current user= {"username":"VTI","role":"VTB-I","priviledges":["productList"],"pgro
ups":["*"],"dbfields":["*"],"queryparms":["*"],"maxcount":"10","iat":1588687317}
10
SELECT * FROM NEDI68.PRODREST ORDER BY PRICESEL desc LIMIT 0, 10
```


[{"PRODID":"340","NAME":"Trockner
XXL","PGROUP":"TROCKNER","PRICEPUR":"300.90","PRICESEL":"749.99","STOCKQTY":"82"},
{"PRODID":"356","NAME":"Trockner
XL","PGROUP":"TROCKNER","PRICEPUR":"270.22","PRICESEL":"639.99","STOCKQTY":"38"},
{"PRODID":"358","NAME":"Trockner L","PGROUP":"TROCKNER","PRICEPUR":"240.00","PRICESEL":"609.99","STOCKQTY":"1"},
{"PRODID":"330","NAME":"Trockner
M","PGROUP":"TROCKNER","PRICEPUR":"220.80","PRICESEL":"549.99","STOCKQTY":"150"},
{"PRODID":"300","NAME":"Trockner S","PGROUP":"TROCKNER","PRICEPUR":"200.30","PRICESEL":"499.99","STOCKQTY":"2"},
{"PRODID":"265","NAME":"Grill 1000","PGROUP":"GRILL","PRICEPUR":"190.80","PRICESEL":"419.99","STOCKQTY":"71"},
{"PRODID":"225","NAME":"Grill 4000S","PGROUP":"GRILL","PRICEPUR":"185.80","PRICESEL":"359.99","STOCKQTY":"37"},
{"PRODID":"220","NAME":"Grill 4000","PGROUP":"GRILL","PRICEPUR":"180.80","PRICESEL":"329.99","STOCKQTY":"94"},
{"PRODID":"210","NAME":"Grill 8000X","PGROUP":"GRILL","PRICEPUR":"175.80","PRICESEL":"289.99","STOCKQTY":"90"},
{"PRODID":"200","NAME":"Grill 8000","PGROUP":"GRILL","PRICEPUR":"170.80","PRICESEL":"269.99","STOCKQTY":"12"}]

POW3R
einfach. digital. agil.

HTTPS für node.js/EXPRESS

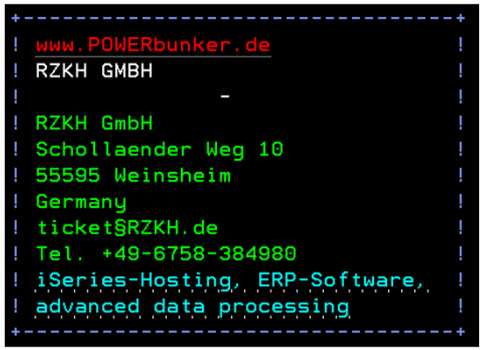
```
var fs = require('fs');
var http = require('http');
var https = require('https');
var privateKey  = fs.readFileSync('sslcert/server.key', 'utf8');
var certificate = fs.readFileSync('sslcert/server.crt', 'utf8');

var credentials = {key: privateKey, cert: certificate};
var express = require('express');
var app = express();

// your express configuration here

var httpServer = http.createServer(app);
var httpsServer = https.createServer(credentials, app);

httpServer.listen(8080);
httpsServer.listen(8443);
```

POW3R
einfach. digital. agil.

REST-Sicherheit – Best Practice

- KIS – Keep it simple
- HTTPS verwenden
- Keine Zugangsdaten in URLs übergeben
  - https://rest-server.com/products?apiKey=abcd999
- Hash für Kennwörter und Tokens
- Verwendung von Rollen
  - Zugriffsvalidierung aus URL-Ebene
  - Freischaltung von Filteroptionen der Datenmenge in der Antwort
- Validierung von Eingabeparametern
  - SQL-Injection verhindern
- Optional: Zeitangabe in Requests
  - Gegen bruteforce mit statischem Request

Vielen Dank an:



POW3R
einfach. digital. agil.

Vielen Dank für Ihr Interesse und Ihre Aufmerksamkeit

Ein Vortrag von:

Dipl. Wirt-Ing.

Dirk Neumann

Dirk.Neumann@Neumann-Software.de
Neumann Softwaretechnik Consulting / Bochum / Germany

www.neumann-software.de

Some people want it to happen,
some wish it would happen,
others make it happen.

(Michael Jordan)

POW3R
einfach. digital. agil.