Schnittstellendokumentation

API DOC

**Autor:** Jiri Valenta

**Datum:** 20.12.2017

**Ort:** Wien

**Applikationname:** API DOC

**Kurs:** Ausbildung zum Web Developer - JavaScript - Diplom-Lehrgang

**Kurskode:** 18294507

**Datei**: Server/server.js

**Zeile 22:** Es wird json Objekt empfangen, geparst und im allprojects.json gespeichert, als Antwort wird Parameter projectcreated geschickt (yes)

**app.post( '/projectsND', function( req, res) {**

console.log( "received data: ", req.body.dataObject );

fs.readFile( "allprojects.json", function(err, data) {

var root={};

root = JSON.parse(data);

root.projects.push(JSON.parse(req.body.dataObject));

res.json({projectcreated: 'yes'});

res.end();

//console.log("root object after insert: ", root);

//persist allprojects in root from to json file

fs.writeFile( "allprojects.json",JSON.stringify(root),function(){

console.log("Projects saved to file");

}); //write file

}); //readFile

}); //app post

**Zeile 40:** Es wird kein Parameter empfangen, als antwort wird Objekt mit allen Projekten( Array ) geschickt. Damit wird Projects combobox refreshed.

app.get('/projectsND', function( req, res){

fs.readFile( 'allprojects.json', function(err, data) {

//create object with id, name detail and put to response

var root={};

var projectsArr=[];

var out={ projects:projectsArr };

root = JSON.parse(data);

//console.log("dataparsed: ",JSON.parse(data));

for(var i=0; i<root.projects.length; i++) {

out.projects[i]={id:i,name:root.projects[i].name,description:root.projects[i].description}

if(root.projects[i].name){

}//out.projects[0]={id:7};

}

res.send( JSON.stringify(out) );

res.end();

}); //read file

});

**Zeile 63:** Es wird id Parameter empfangen, als antwort wird ein Objekt mit dem aktuellen Projekt geschickt. Damit wird Detail von Projekt aufgebaut.

app.get('/projectsND/:id', function( req, res){

var id = req.params.id;

fs.readFile( 'allprojects.json', function(err, data) {

//create object with id, name detail and put to response

var root={};

var out={};

root = JSON.parse(data);

out=root.projects[id];

res.send( JSON.stringify(out) );

res.end();

}); //read file

});

**Zeile 82:** Es wird json Objekt von Projekt empfangen, geparst und im allprojects.json auf der Stelle mit empfangenem id gespeichert. Als Anwort wird JSON parameter updated geschickt

app.put('/projectsND/:id', function(req, res) {

console.log("update activated");

var id = req.params.id;

console.log("id to update: " + id);

var receivedProject = JSON.parse(req.body.dataObject);

fs.readFile( "allprojects.json", function(err, data) {

var root={};

root = JSON.parse(data);

console.log("id before update change", id);

root.projects[id]=JSON.parse(JSON.stringify(receivedProject))

//persist allprojects in root from to json file

fs.writeFile( "allprojects.json",JSON.stringify(root),function(){

res.json({updated: root});

res.end();

console.log("Project updated");

}); //write file

}); //readFile

//res.send('Successfully deleted product!');

});

**Zeile 104:** Es wird id Parameter empfangen, der Array Position von zu Löschen gespeichertem Projekt zeigt

app.delete('/projectsND/:id', function(req, res) {

console.log("delete activated");

var id = req.params.id;

console.log("id to delete: " + id);

fs.readFile( "allprojects.json", function(err, data) {

var root={};

root = JSON.parse(data);

root.projects.splice(id,1);

//persist allprojects in root from to json file

fs.writeFile( "allprojects.json",JSON.stringify(root),function(){

//res.json({deleted: 'yes'});

res.end();

console.log("Project deleted");

}); //write file

}); //readFile

});

**Zeile 125:** Es wird id Parameter empfangen, der Array Position von zu Löschen gespeicherter Connection erhält. Ein id von entsprechendem id von Projekt wird auch empfangen. Connection wird gelöscht. Als Antwort wird connectionDeleted JSON Parameter geschickt.

app.post('/projectsND/:id', function(req, res) {

console.log("delete of connection activated");

var projectID = req.params.id; //connection id

var connectionID =JSON.parse(req.body.conID);

console.log( "received projectID: ", projectID );

console.log("connection id to delete: " + connectionID);

fs.readFile( "allprojects.json", function(err, data) {

var root={};

root = JSON.parse(data);

console.log( "received projectID: ", projectID );

console.log("connectionID bevore splice: ",connectionID );

root.projects[projectID].connections.splice(connectionID,1);

//persist allprojects in root from to json file

fs.writeFile( "allprojects.json",JSON.stringify(root),function(){

res.json({connectionDeleted: connectionID});

res.end();

console.log("Project deleted",connectionID);

}); //write file

}); //readFile

});