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 der Parameter projectcreated geschickt (yes)

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

console.log( "receiveddata: ", 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);

//persistallprojects in root fromtojsonfile

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

console.log("Projects savedtofile");

}); //writefile

}); //readFile

}); //apppost

**Zeile 40:** Es wird kein Parameter empfangen, als Antwort wird ein Objekt mit allen Projekten( Array ) geschickt. Damit wird Project‘s combobox refreshed.

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

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

//createobjectwithid, namedetail and puttoresponse

var root={};

varprojectsArr=[];

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();

}); //readfile

});

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

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

varid = req.params.id;

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

//createobjectwithid, namedetail and puttoresponse

var root={};

var out={};

root = JSON.parse(data);

out=root.projects[id];

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

res.end();

}); //readfile

});

**Zeile 82:** Es wird json Objekt vom Projekt empfangen, geparst und im allprojects.json auf der Stelle mit empfangener id gespeichert. Als Antwort wird JSON parameterupdated geschickt.

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

console.log("update activated");

varid = req.params.id;

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

varreceivedProject = JSON.parse(req.body.dataObject);

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

var root={};

root = JSON.parse(data);

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

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

//persistallprojects in root fromtojsonfile

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

res.json({updated: root});

res.end();

console.log("Project updated");

}); //writefile

}); //readFile

//res.send('Successfullydeletedproduct!');

});

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

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

console.log("deleteactivated");

varid = req.params.id;

console.log("idtodelete: " + id);

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

var root={};

root = JSON.parse(data);

root.projects.splice(id,1);

//persistallprojects in root fromtojsonfile

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

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

res.end();

console.log("Project deleted");

}); //writefile

}); //readFile

});

**Zeile 125:** Es wird ein id Parameter empfangen, der die Array Position von der zu Löschen gespeicherten Connection erhält. Eine id von entsprechender id vom Projekt wird auch empfangen. Die Connection wird gelöscht. Als Antwort wird ein connectionDeleted JSON Parameter geschickt.

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

console.log("deleteofconnectionactivated");

varprojectID = req.params.id; //connectionid

varconnectionID =JSON.parse(req.body.conID);

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

console.log("connectionidtodelete: " + connectionID);

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

var root={};

root = JSON.parse(data);

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

console.log("connectionIDbevoresplice: ",connectionID );

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

//persistallprojects in root fromtojsonfile

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

res.json({connectionDeleted: connectionID});

res.end();

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

}); //writefile

}); //readFile

});