**Node.js Workshop: Node events (EventEmitter)**

After completing this workshop the student is knows how to:

* setting up development environment
* build a customer log module
* how to install events module and EventEmitter
* how to complete the logEvents project

**What are Node.js events**

When you worked with JavaScript in the browser, you know how much of the interaction of the user is handled through events: mouse clicks, keyboard button presses, reacting to mouse movements, and so on. On the backend side, Node.js offers us the option to build a similar system using the events module.

Official [Web page](https://nodejs.org/en/learn/asynchronous-work/the-nodejs-event-emitter).

events [documentation](https://nodejs.org/api/events.html)

**Exercise 1. index.js and initialize your events project**

1. Create a new folder for these assignments. Save all your code in there.
2. Create a new file and name it to *index.js.* Make your new server to log *Hello world* in console. Then run your new server and make sure it works
3. Next you need to initialize your project by typing *npm init* in your terminal.
4. Modify your package.json file so that in “scripts” your server start with *node index.js* and in development mode you can start your server with *nodemon index.js*
5. Create your first event handler server as follows – you need to fill in the missing parts (…?)

// require the events module

const EventEmitter = …?

// create an instance of EventEmitter

const eventEmitter = …?

// register a listener for the start event

eventEmitter.on('start', () => {

    console.log('started');

});

// emit the start event

eventEmitter.emit('…?');

**Exercise 2. Create a Log events register project**

1. Create a new folder for these assignments. Save all your code in there.
2. Initialize your project with *npm init* in your terminal
3. Install required date-fns and uuid modules to your project (*npm install <module\_name>*) and nodemon module for development (parameter -D)
4. Update project package.json file so that the start script is “*node index.js*” and dev script is “*nodemon index.js*”
5. Create a file to *logEvents.js* module *-* you need to fill in the missing parts (…?)

// This module logs events to a file

// require the date-fns module

const { format } = …?

// require the uuid module

const { v4: uuid } = …?

// require the file system module

const fs = …?

// require the file system promises module

const fsPromises = require('fs').promises;

// require the path module

const path = …?

// this async function logs the message to a file

const logEvents = async (message) => {

    const dateTime = `${format(new Date(), 'dd-MM-yyyy HH:mm:ss')}`;

    //this is the log item to be written to the file

    const logItem = `${dateTime}\t ${uuid()}\t ${message}\n`;

    console.log(logItem);

    try {

        // check if the logs directory exists

        if (!fs.existsSync(path.join(\_\_dirname, 'logs'))) {

            // if not create the directory

            await fsPromises.mkdir(path.join(\_\_dirname, 'logs'));

        }

        // write the log item to the file

        await fsPromises.appendFile(path.join(\_\_dirname, 'logs', 'eventLog.txt'), logItem);

    } catch (error) {

        // log any errors

        console.error(`Errori  writing to log: ${error}`);

    }

}

// export the logEvents function

module.exports = …?;

1. Next you need to create *index.js* file *-* you need to fill in the missing parts (…?)

// Description: This file is used to test the event emitter.

// require the logEvents module you created in previous step

const logEvents = …?

// require the events module

const eventEmitter = …?

// create a class that extends the event emitter class

class MyEmitter extends eventEmitter { }

// initialize the event emitter

const myEmitter = new MyEmitter();

// add listener for the log event and call the logEvents function

myEmitter.on('log', (msg) => …?);

// set a timeout to emit the log event

setTimeout(() => {

    // emit the log event

    myEmitter.emit('log', 'This is a test message2');

}, 1000);

1. Test you logEvents (“*npm run dev*”) project – ensure there is a log directory and eventLog.txt file which looks somewhat like this…

