

# Data storage

@somkiat.cc



# Store data

In-memory

File system

Relational database

Non-relational database



# In-memory



# Counter

```
var http = require('http');  
var counter = 0;  
  
var server = http.createServer(function(req, res) {  
  counter++;  
  res.write('I have been accessed ' + counter + ' times.');
```

```
  res.end();  
}).listen(8888);
```



# Counter

\$node counter.js



I have been accessed 1 times.

I have been accessed 3 times.

I have been accessed 5 times.



# Counter

```
var http = require('http');
var counter = 0;

var server = http.createServer(function(req, res) {
  if(req.url === '/favicon.ico'){
    console.log('favicon');
    return;
  }
  counter++;
  res.write('I have been accessed ' + counter + ' times. ');
  res.end();
}).listen(8888);
```



# File system



# Todo list with command line

List of tasks

Add new task





# Todo list with command line

```
$node todo.js list
```

```
$node todo.js add <task>
```



# Step 1 :: command line

`$node todo.js` **add learn node.js**

```
var fs = require('fs')
var path = require('path')
var args = process.argv.splice(2)
var action = args.shift()
var newTask = args.join(' ')
```

```
console.log(action)
console.log(newTask)
```



# Step 2 :: action/command

```
var fs = require('fs')
var path = require('path')
var args = process.argv.splice(2)
var action = args.shift()
var newTask = args.join(' ')

switch (action) {
  case 'list':
    listTasks()
    break;
  case 'add':
    addTask(newTask)
    break;
  default:
    console.log('please use addlist')
}
```



# Step 3 :: Add new task

```
const DB_FILE_LOCATION = 'demo.txt'
```

```
var writeDataToFile = (tasks) => {  
  fs.writeFile(DB_FILE_LOCATION, JSON.stringify(tasks), 'utf8',  
(error) => {  
    if(error) throw error  
    console.log('New task is saved.')  
  })  
}
```

```
var addTask = (newTask) => {  
  readDataFromFile((tasks) => {  
    tasks.push(newTask)  
    writeDataToFile(tasks)  
  })  
}
```



# Step 4 :: List all task

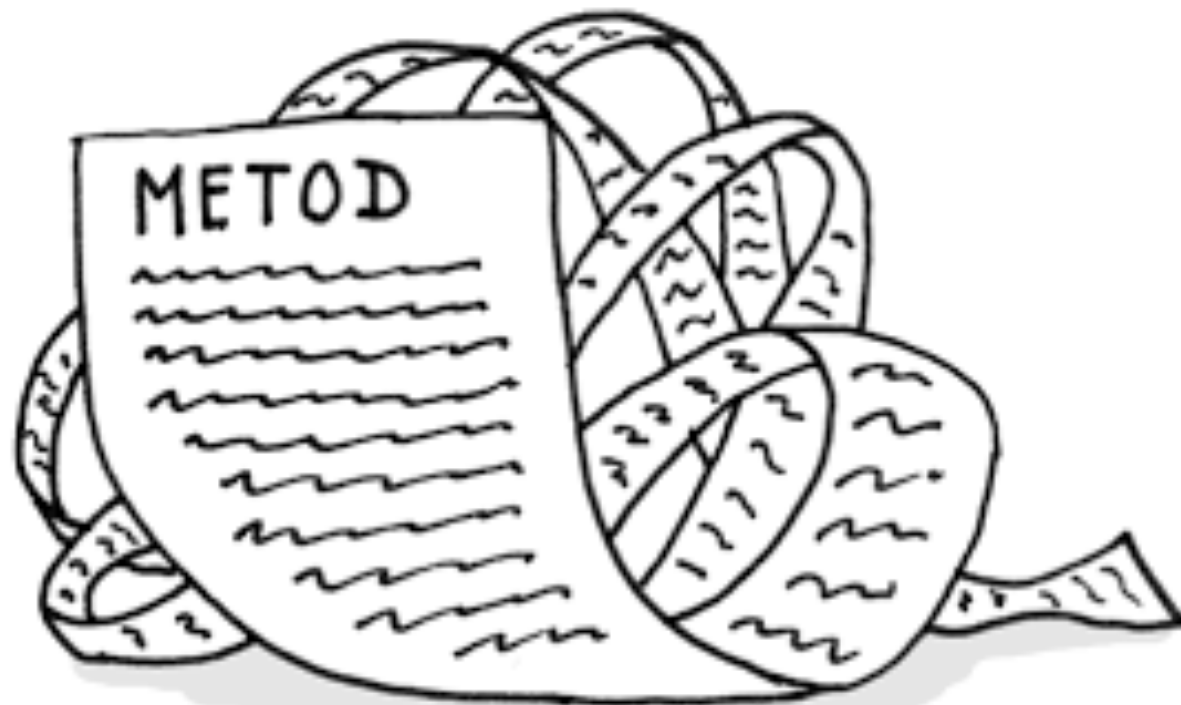
```
var readDataFromFile = (callback) => {  
  fs.exists(DB_FILE_LOCATION, (exists) => {  
    var tasks = []  
    if(exists) {  
      fs.readFile(DB_FILE_LOCATION, 'utf8', (error, data) => {  
        if(error) throw error  
        var task = data.toString()  
        var tasks = JSON.parse(task || '[]')  
        callback(tasks)  
      })  
    } else {  
      callback([])  
    }  
  })  
}
```



# Step 4 :: List all task

```
var listTasks = () => {  
    readDataFromFile((tasks) => {  
        for (var index in tasks) {  
            console.log(tasks[index])  
        }  
    })  
}
```





# Demo :: Work with modular

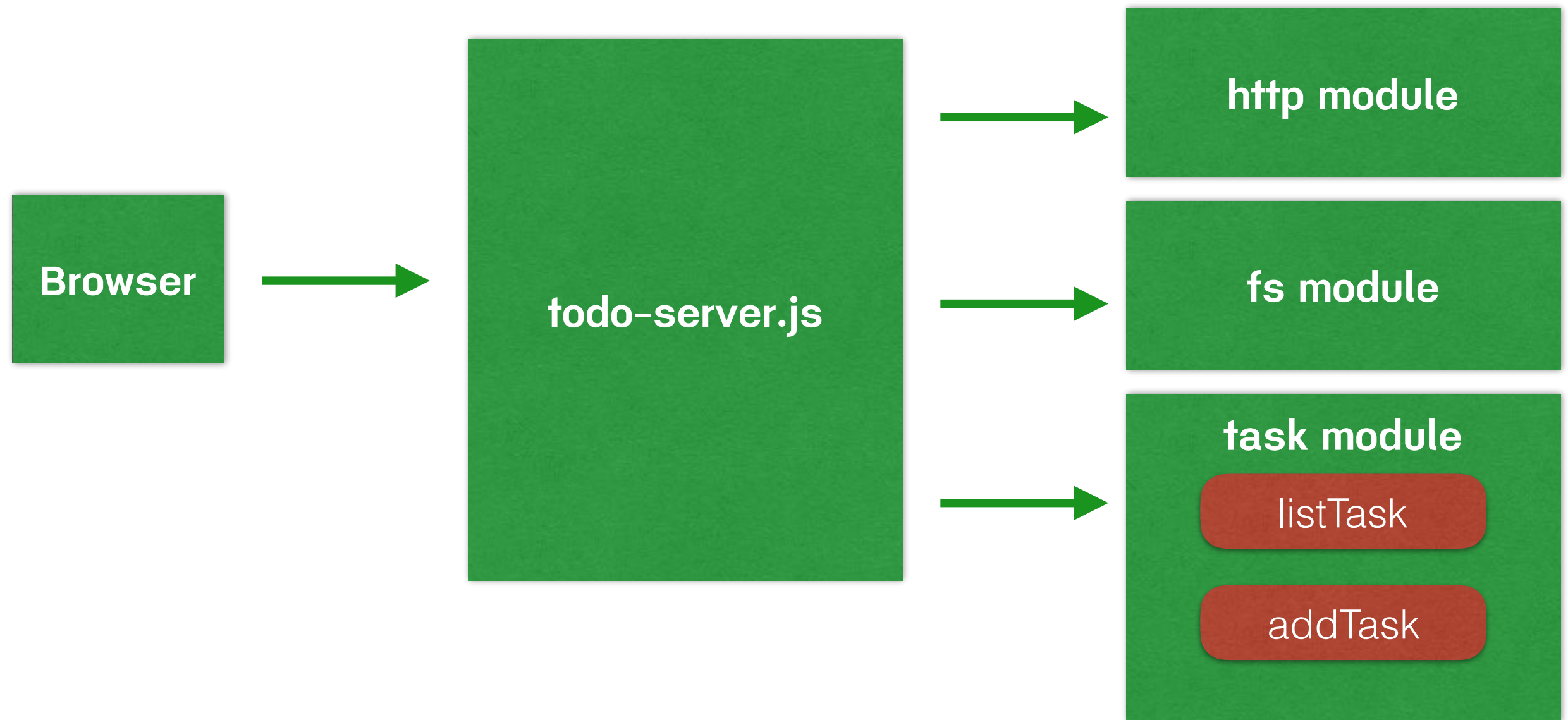




# Workshop



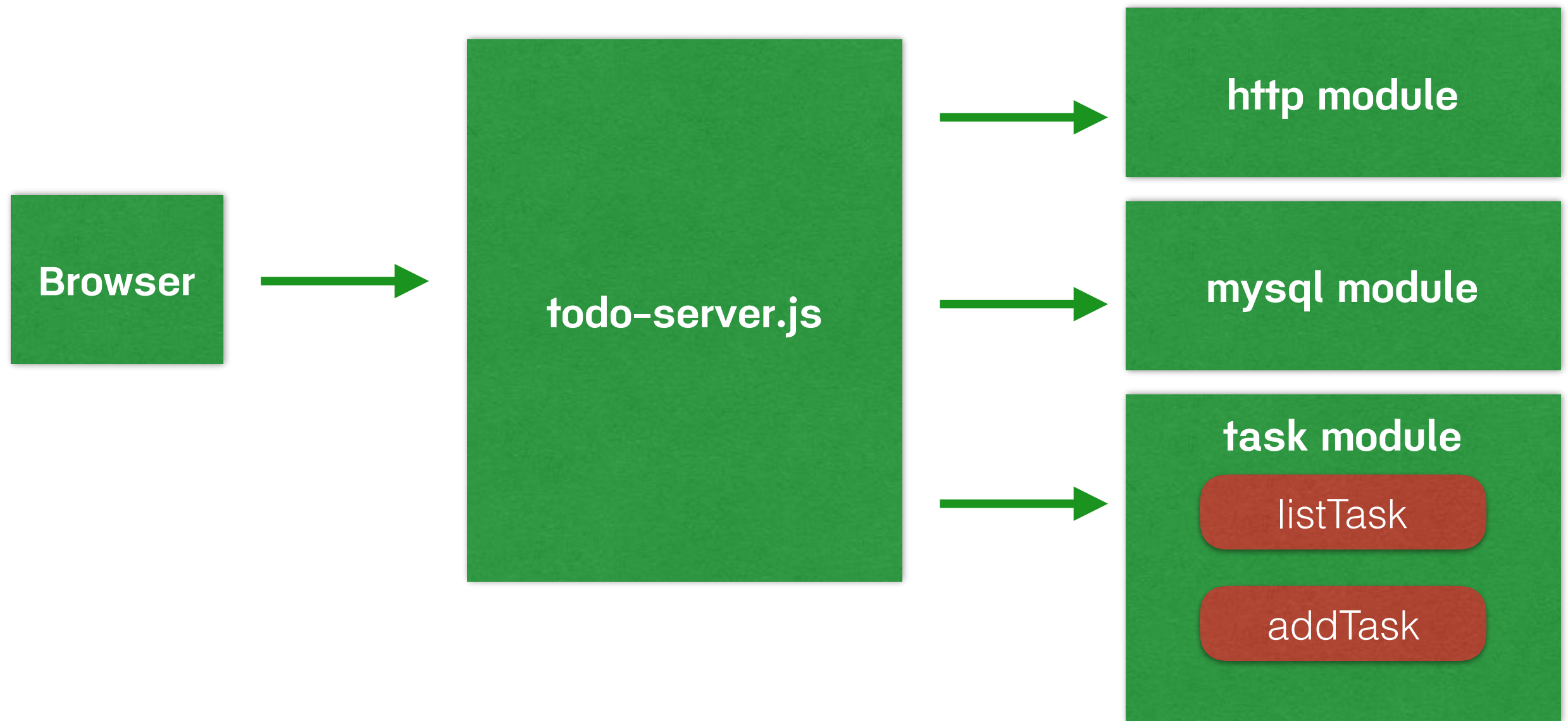
# Web app with Todo list



# Relational database



# Structure



# Non-Relational database



# Structure

