



Übung01 NodeJS

Roland Paltz



NodeJS?

- Serverseitige JavaScript-Laufzeitumgebung
- Basierend auf Google Chromes V8-Engine
- Eventorientierte Architektur
- Modular aufgebaut (npm)
- Optimal für Serveranwendungen mit:
 - Hohem Datendurchsatz
 - Hoher Anzahl gleichzeitiger Verbindungen
 - Geringem Rechenaufwand



EXKURS: JavaScript

- > Scriptsprache
- > Dynamisch typisiert
- > Objektorientiert
- > NICHT klassenorientiert

Bsp: UDP-Socket

```
var socket = require('dgram').createSocket('udp4');
```

```
socket.on('message', function (msgbuffer, rinfo) {
```

```
  console.log(msgbuffer.toString());
```

```
});
```

```
socket.bind(27017);
```

Warum JSON?

➔ JSON = JavaScript Object Notation

```
{  
  foo: 1,  
  bar: "something",  
  arr: [  
    1,  
    "something",  
    {  
      hello: "world"  
    }  
  ]  
}
```



```
{  
  "foo": 1,  
  "bar": "something",  
  "arr": [  
    1,  
    "something",  
    {  
      "hello": "world"  
    }  
  ]  
}
```

Aufbau einer Nachricht

```
{  
  "from": "ID of the sender",  
  "type": "Type of the message: [APP | CONTROL]",  
  "action": "Action of the message:[RUMOR, KILL, etc.]",  
  "msg": "Msg-text or content"  
}
```

Ablauf der Anwendung

- Input-Dateien einlesen/parsen
- Socket definieren und Events binden
- Socket an Port binden
- Verarbeiten der Nachrichten im `message`-Event (nach type und action)
- Send-Funktion zum senden von Nachrichtenobjekten
- Close-Funktion zum schließen des Sockets
- Senden von CONTROL-Nachrichten über 2. Anwendung



Vorführung





Noch Fragen?





Vielen Dank für eure
Aufmerksamkeit