

# Veebiarendus

## Back-End arendus

Martti Raavel

[martti.raavel@tlu.ee](mailto:martti.raavel@tlu.ee)

# Tänaõsed teemad

- Meenutame eelmist loengut
- [Middleware](#)
- [Express Router](#)

**Millest rääkisime eelmises loengus?**

## Vahevara ehk **Middleware**

**Middleware** funktsioonid on funktsioonid, millel on juurdepääs päringuobjektile ( **req** ), vastuseobjektile ( **res** ) ja järgmisele funktsioonile rakenduse päringu-vastuse tsüklis.

## Middleware

Põhimõtteliselt võime `middleware` 'i mõelda kui filtrist, mis töötleb päringuid enne, kui need jõuavad päringu-vastuse tsükli järgmisse etappi.

## Middleware kasutamine

Middleware saab:

- Käivitada koodi
- Teha muudatusi request ja response objektides
- Lõpetada request-response tsüklit
- Kutsuda välja järjekorrast järgmine middleware
- ...

## Middleware registreerimine

Middleware -t saab rakendada:

- kõikidele päringutele
- ainult teatud marsuutidele
- ...

## Next funktsioon

Next funktsioon on Express-ruuteri funktsioon, mis käivitamisel käivitab middleware praeguse middleware 'i järel.



## Middleware näide

```
// Http päringute konsooli logimise middleware

const logger = (req, res, next) => {
  // Väljastatakse päringu sihtaadress, meetod ja aeg
  console.log(req.url, req.method, new Date().toISOString());
  // Next funktsiooni käivitamine annab järjekorra üle järgmisele middleware'le
  return next();
}
```

## Middleware näite registreerimine kõikidele päringutele

```
...
```

```
// Middleware registreerimine  
app.use(logger);
```

```
...
```

## Middleware näite registreerimine ainult teatud marsuutidele

```
...  
  
// Middleware registreerimine  
app.get('/api', logger, (req, res) => {  
  res.send('Hello World!');  
});  
  
...
```

## Not Found middleware

Üks võimalikke kasutuskohi `middleware` 'le on 404 lehekülje kuvamine, kui päringu sihtressurssi ei leita.

## Not Found middleware näide

```
// 404 lehekülje kuvamine, kui päringu sihtressurssi ei leita

const notFound = (req, res, next) => {
  res.status(404).send({
    success: false,
    message: 'Not Found'
  });
}
```

## Not Found middleware registreerimine

```
// Middleware importimine
const notFound = require('./middlewares/notFound');

...

// Middleware registreerimine
app.use('*', notFound);

...
```

# Express Router

`Router` on `Express` raamistiku objekt, mis võimaldab meil marsruute grupeerida.

## Express Router kasutamine

Express Routeri kasutamiseks loome kõigepealt iga ressursi kohta eraldi faili, kuhu kirjutame vastava ressursi marsruudid.



# Express Router näide

```
// autodega seotud marsruudid

const express = require('express');
const router = express.Router();

const carsControllers = require('./carsControllers');

router.get('/', carsControllers.getAll);
router.get('/:id', carsControllers.getById);
router.post('/', carsControllers.create);

module.exports = router;
```

# Express Router importimine ja kasutamine

`Router` objekti saab kasutada `Express` rakenduses, kasutades `app.use` meetodit.

```
// Router importimine

const carsRoutes = require('./cars/carsRoutes');

...

// Router kasutamine

app.use('/cars', carsRoutes);

...
```

# Kodune töö

- Loe läbi tänase loengu materjalid:
  - [Middleware](#)
  - [Express Router](#) materjalid
- Struktureeri oma projekti marsruudid `Router` objektideks
- Lisa oma projekti `middleware` funktsioon, mis ...