



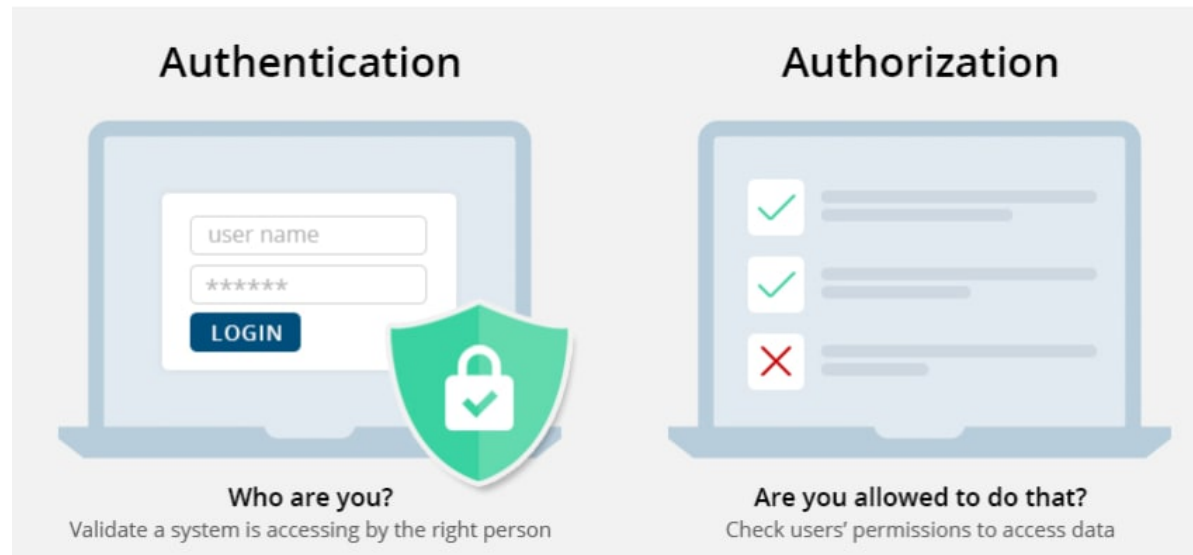
# Form Validation

# Today's topics

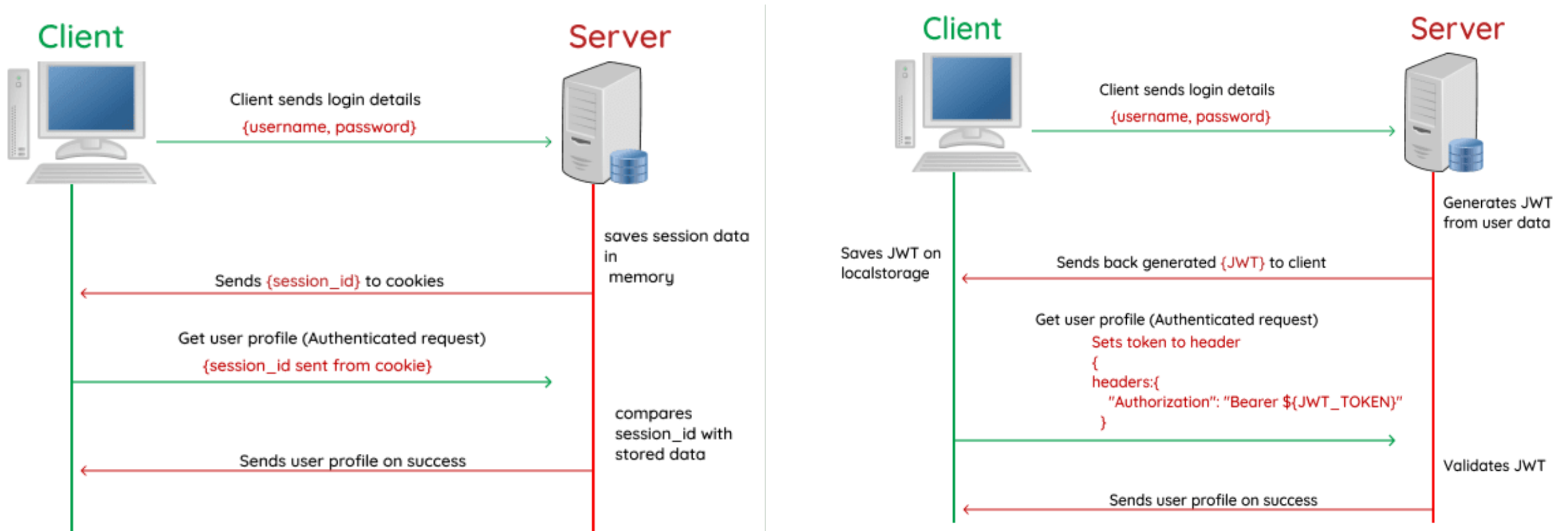
- Overview of Authentication System
- Backend Form Validation with Joi
  - Example: Registration API
- Frontend Form Validation with Vuelidate
  - Example: Registration Page

# Authentication vs Authorization

- **Authentication** confirms that users are who they say they are.
- **Authorization** Verifying user's permissions to access resources.



# Session Based vs Token Based Authentication



# Session Based vs Token Based Authentication

## Sesstion Based

- User state is store in server (memory/database)
- Use Cookie to store session-id

NPM **express-session**

<https://www.npmjs.com/package/express-session>

## Token Based

- User state is stored on client (localStorage)
- Attatch token inside request header

NPM **express-jwt**

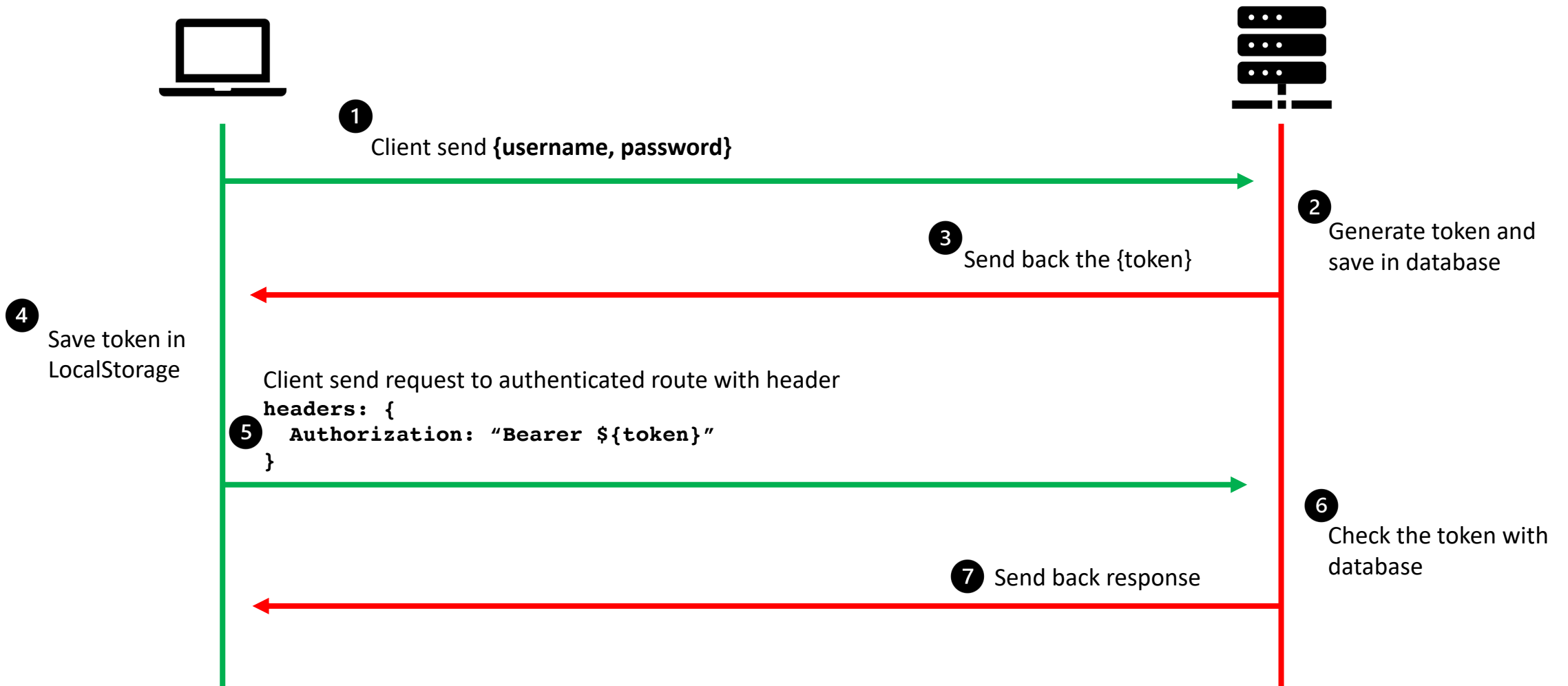
<https://www.npmjs.com/package/express-jwt>



# Simple Token Authentication

Let's re-invent the wheel

# Example (Simple Token Authentication)



# Example (Simple Token Authentication)

## Database

- User Table
  - Id
  - Username
  - Password (encrypted)
  - Email
  - Mobile
- Token Table
  - FK User Id
  - token

## API

- /api/user/register/  
(With Form Validation)
- /api/user/login/
- /api/user/me/

## Pages

- Register Page
- Login Page
- Home Page (require auth)



# Starting Code

- Explain SQL files
- Explain signup.html

# Backend Form Validation with Joi



```
1 const express = require('express')
2 const Joi = require('joi')
3
4 app = express()
5
6 const schema = Joi.object({
7   name: Joi.string().required(),
8   age: Joi.number().required().min(0)
9 })
10
11 app.get('/', (req, res, next) => {
12   try {
13     schema.validateAsync(req.query)
14   } catch (err) {
15     res.status(400).send(err)
16   }
17
18   res.send('ok')
19 })
```

<https://joi.dev/>

//TODO Demo this example  
with postman

# Backend Form Validation with Joi

## Joi API

- [Joi.string\(\)](#)
- [Joi.number\(\)](#)
- [Joi.date\(\)](#)
- [Joi.any.custom\(\)](#)
- [Etc.](#)

// TODO Demo the example with postman  
// TODO Implement Register API

# Frontend Validation with Vuelidate

- Why Frontend Validation
  - Instant feedback
  - Reduce server load
  - Improve user experience
- Why Backend Validation
  - Frontend validation can be bypass
  - Request might not come from trusted client
  - Backend Validation is a **MUST!!!** It is the only way to ensure that incoming data is valid

Garbage in, garbage out

# Frontend Validation with Vuelidate

## Installation

```
<script src="vuelidate/dist/vuelidate.min.js"></script>
```

Html

```
// global
Vue.use(window.vuelidate.default)

// local mixin
var validationMixin = window.vuelidate.validationMixin
```

JavaScript

<https://vuelidate.js.org/#sub-basic-form>

//TODO demo the example on vuelidate.js.org

# Frontend Validation with Vuelidate

## API

- [Builtin Validator](#)
- [Custom Validator](#)

# Exercises

- Add Joi validation to other API
- Add Vuelidate to other page