

TypeScript ja Vue

1. Ülevaade kasutatavast tehnoloogiast

Mobiilirakenduste arendamiseks kasutame **TypeScripti**, **Vue.js-i**, **Ionicut** ja **Capacitorit**. Need tehnoloogiad võimaldavad meil luua ühe koodibaasiga rakendusi, mis töötavad nii iOS-i kui ka Androidi seadmetes.

TypeScript

TypeScript on **JavaScripti laiendus**, mis lisab staatilise tüüpimise. See tähendab, et erinevalt JavaScriptist, kus muutujate tüüpe kontrollitakse alles koodi käivitamisel (runtime), võimaldab TypeScript neid tüüpe määrata juba koodi kirjutamise ajal.

TypeScripti eelised:

- Staatiline tüüpimine vähendab vigade tekkimise võimalust.
- Koodi parem loetavus ja hooldatavus.
- Suurem koostöövõime suurte projektide puhul.
- Parem integreeritus arenduskeskkondadesse nagu VS Code.

Vue.js

Vue.js on **progressiivne JavaScripti raamistik**, mida kasutatakse kasutajaliideste loomiseks. Vue töötab komponendipõhiselt ja võimaldab lihtsat reaktiivset andmete sidumist. See on eriti kasulik mobiilirakenduste puhul, kus dünaamilisus ja kiirus on olulised.

Vue.js peamised omadused:

- **Komponentidel põhinev arhitektuur** – lihtsustab koodi organiseerimist ja korduvkasutust.
- **Reaktiivne andmete haldus** – muudab kasutajaliidese automaatselt, kui andmed muutuvad.

- **Hea integreeritus TypeScriptiga** – võimaldab tugevat tüübituge Vue rakenduste loomiseks.

Ionic

Ionic on **hübriidrakenduste raamistik**, mis võimaldab luua mobiilirakendusi kasutades veebitehnoloogiaid (HTML, CSS, JavaScript/TypeScript). Ionic pakub mobiili jaoks optimeeritud UI-komponente ja integreerub hästi Vue ning TypeScriptiga.

Ionicu peamised omadused:

- **Kasutajaliidese komponendid** – mobiilseadmetele optimeeritud UI elemendid (nt `ion-button`, `ion-list`, `ion-toast`).
- **Tugev integratsioon mobiiliplatvormidega** – töötab koos Capacitoriga, et pakkuda ligipääsu natiivsetele funktsioonidele.
- **Ühe koodibaasiga mitmele platvormile** – iOS ja Android ühest koodibaasist.

Capacitor

Capacitor on **natiivsete funktsioonide integreerimise kiht**, mis võimaldab Ionicu rakendustel kasutada mobiiliseadmete riistvara nagu kaamera, failisüsteem ja GPS. See on alternatiiv Cordovale, pakkudes moodsamat ja kiiremat API-d.

Capacitori peamised omadused:

- **Ligipääs natiivsetele API-dele** – kaamera, failisüsteem, teavitused, GPS jne.
- **Modernne ja kiire** – toetab kaasaegseid arendusstandardeid.
- **Paindlikkus** – võimaldab lisada ka kohandatud natiivkoodi, kui vaja.

2. TypeScripti erisused võrreldes JavaScriptiga

TypeScript toob kaasa mitmeid täiustusi, mis eristavad seda tavalisest JavaScriptist:

Tüüpide kasutamine

JavaScriptis võivad muutujad olla dünaamilised, kuid TypeScriptis saab need defineerida täpselt:

```
let count: number = 10;
let username: string = "Kasutaja";
let isActive: boolean = true;
```

See aitab vältida vigu, kus näiteks numbrit üritatakse käsitleda tekstina.

Funktsioon

```
function sum(a: number, b: number): number {
    return a + b;
}
```

Generics

Generics võimaldab luua korduv kasutatavaid funktsioone ja klasse:

```
function identity<Type>(arg: Type): Type {
    return arg;
}
```

Liideste kasutamine

TypeScript võimaldab luua liideseid, mis määratlevad objektide struktuuri:

```
interface User {
    id: number;
    name: string;
    email: string;
}
```

Seda kasutatakse sageli API vastuste ja andmemudelite defineerimiseks.

Klassid ja abstraktsioonid

TypeScript toetab objektorienteeritud programmeerimist, mis teeb koodi korduvkasutatavamaks:

```
class Animal {  
    name: string;  
  
    constructor(name: string) {  
        this.name = name;  
    }  
  
    makeSound(): void {  
        console.log("Hääl!");  
    }  
}
```

Sellised struktuurid muudavad rakenduste arhitektuuri paremini hallatavaks.

3. Vue koos TypeScriptiga

Vue.js toetab TypeScripti kasutamist, et muuta komponendid paremini hallatavaks.

Näide Vue komponendist TypeScriptiga:

```
<script lang="ts">
import { defineComponent } from 'vue';

export default defineComponent({
  data() {
    return {
      count: 0 as number,
    };
  },
  methods: {
    increment(): void {
      this.count++;
    },
  },
});
</script>

<template>
  <button @click="increment">Click me {{ count }}</button>
</template>
```

TypeScripti kasutamine aitab vältida runtime vigu ja muudab komponendid paremini hooldatavaks.

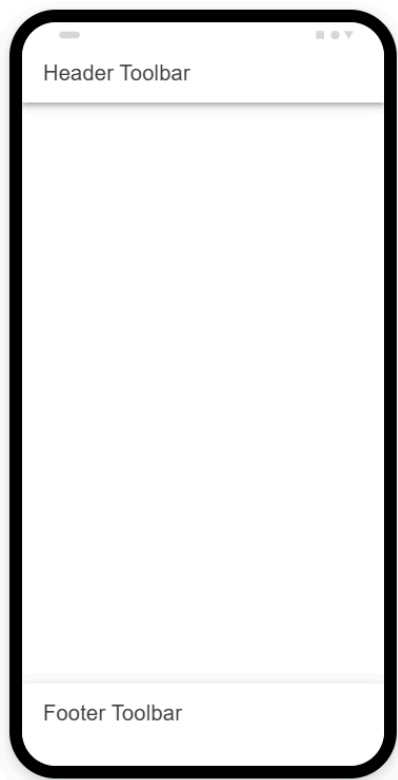
4. Ionicu ja Capacitori eripärad

Ionicu UI komponendid

Ionic pakub mobiilisõbralikke UI-komponente, mida tavaliselt Vue veebiarenduses ei kasutata.

```
<template>
  <ion-header>
    <ion-toolbar>
      <ion-title>Header Toolbar</ion-title>
    </ion-toolbar>
  </ion-header>

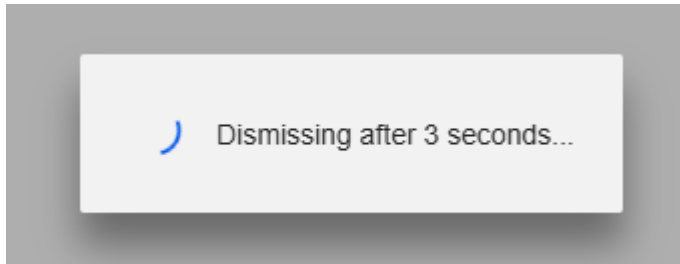
  <ion-footer>
    <ion-toolbar>
      <ion-title>Footer Toolbar</ion-title>
    </ion-toolbar>
  </ion-footer>
</template>
```



Ionicu laadimisindikaator (loading)

See funktsioon kuvab laadimisakna, mis sulgub automaatselt 3 sekundi pärast.

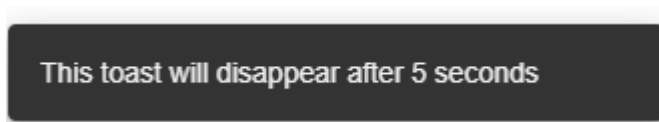
```
<template>
  <ion-button id="open-loading">Show Loading</ion-button>
  <ion-loading trigger="open-loading" :duration="3000" message="Dismissing
after 3 seconds..."> </ion-loading>
</template>
```



Ionicu teavitused (toast)

See koodinäide kuvab ekraanil lühiajalise teavituse.

```
<ion-button id="open-toast" expand="block">Open</ion-button>
  <ion-toast trigger="open-toast" message="This toast will disappear after
5 seconds" :duration="5000"></ion-toast>
```



Natiivsete funktsioonide kasutamine Capacitoriga

Kaamera kasutamine

See funktsioon avab seadme kaamera ja tagastab pildi URL-i.

```
import { Camera, CameraResultType } from '@capacitor/camera';

async function takePhoto() {
  const image = await Camera.getPhoto({
    quality: 90,
    resultType: CameraResultType.Uri,
  });

  console.log(image.webPath);
}
```

Failide salvestamine seadmesse

See funktsioon salvestab tekstifaili seadme mälusse.

```
import { FileSystem, Directory, Encoding } from '@capacitor/filesystem';

async function saveFile() {
  await FileSystem.writeFile({
    path: 'test.txt',
    data: 'Tere, see on testfail!',
    directory: Directory.Documents,
    encoding: Encoding.UTF8
  });
}
```

Geolokatsiooni kasutamine

See funktsioon hangib seadme hetke GPS-koordinaadid.

```
import { Geolocation } from '@capacitor/geolocation';

async function getCurrentPosition() {
  const coordinates = await Geolocation.getCurrentPosition();
  console.log('Latitude:', coordinates.coords.latitude);
  console.log('Longitude:', coordinates.coords.longitude);
}
```


5. Kokkuvõte – kuidas erineb tavapärasest arendusest?

- **TypeScript** lisab range tüübihalduse ja parema veakontrolli võrreldes tavalise JavaScriptiga.
- **Vue koos Ionicuga** võimaldab luua mobiilirakendusi, mida varem arendati näiteks React Native või NativeScriptiga.
- **Capacitor pakub natiivsete funktsioonide tuge**, mida veebirakendustes üldiselt ei kasutata.
- **Mobiiliarenduses tuleb arvestada riistvaralise ligipääsuga**, näiteks kaamera ja GPS, mida Vue veebirakendustes ei kohta.

Kui oled harjunud arendama ainult Vue.js veebirakendusi, siis mobiiliarenduses tuleb rohkem tähelepanu pöörata **natiivsete funktsioonide integreerimisele, staatilisele tüüpimisele ja optimeeritud UI komponentide kasutamisele**.