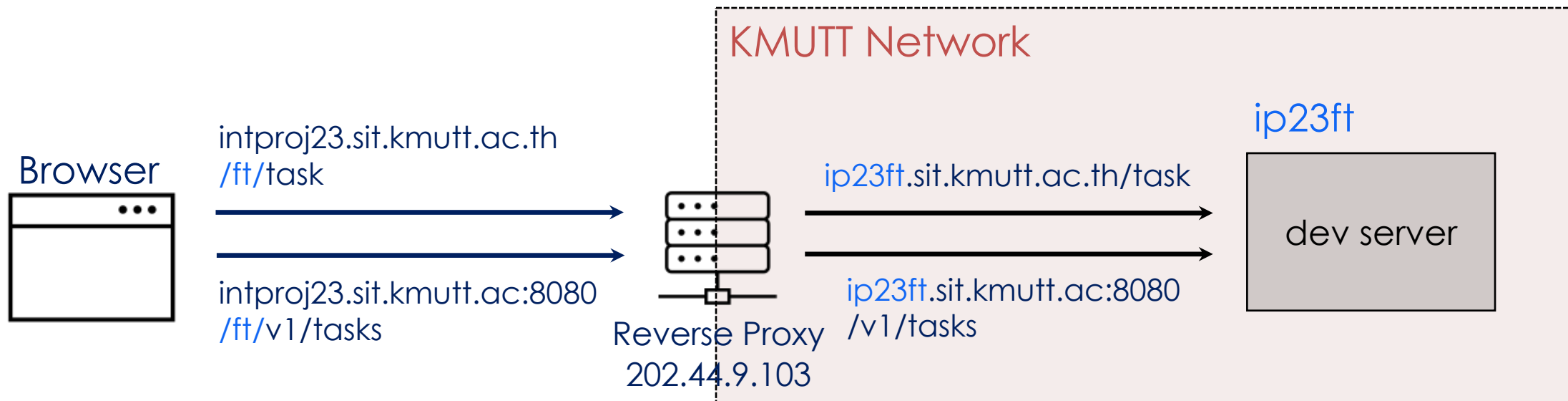


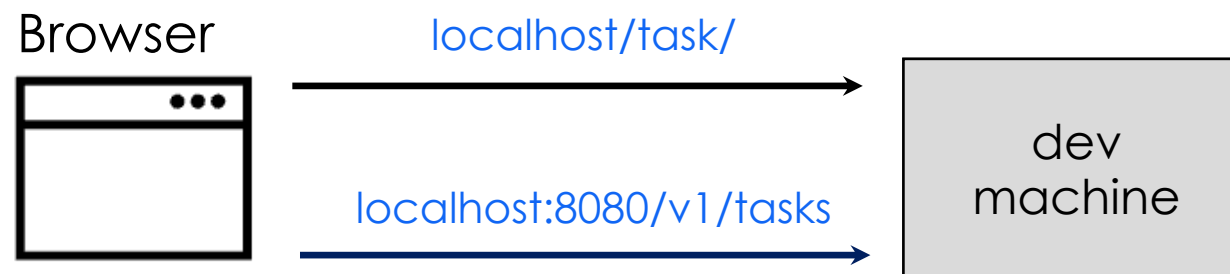
GUIDE 3

Nested Public Path

1. สามารถเข้าถึง ITB-KK ผ่าน URL <http://intproj23.sit.kmutt.ac.th/<TEAMCODE>/>
2. ไม่ hard code ค่า Nested Path (TEAMCODE)
3. สามารถพัฒนาและทดสอบบนเครื่องผู้พัฒนาได้ (ไม่มี Nested Path)
4. (New) รองรับการเรียกใช้ทั้งผ่าน public url และ private url (ต้องติดตั้ง reverse proxy)



Reverse Proxy จะ forward port 80, 443, 8080 ไปที่ VM โดยดูจาก nested public path และจะไม่ส่ง nested path นี้ต่อไปที่ VM



1. ใช้ Vite ที่ เป็น build tool ในการเปลี่ยนค่า URL ของ files และ pages ตามค่า **base** ใน production mode เมื่อ deploy ขึ้น dev server (**npm run build**)

package.json

```
"scripts": {  
  "dev": "vite",  
  "build": "vite build --base=/ft/",  
  "preview": "vite preview"  
},
```

กำหนดค่า **base** เป็น <TEAMCODE> เช่น **/or1/**
เมื่อ build แล้วจะได้ ./dist/index.html ดังรูป

dist/index.html

```
<script type="module" crossorigin src="/ft/assets/index-BwbRPgEm.js"></script>  
<link rel="stylesheet" crossorigin href="/ft/assets/index-BvTfU54A.css">
```

2. กำหนดค่า **base** ใน `createWebHistory(base?)` เพื่อให้ router รู้ค่า **base** ที่กำหนดไว้
ตอน build ทำให้เราไม่ต้องใส่ค่า **base** ไว้ใน path

src/router/index.js

```
const router = createRouter({
  history: createWebHistory(import.meta.env.BASE_URL),
  routes: [
    {
      path: '/',
      redirect: '/task'
    },
    {
      path: '/task',
      name: 'taskList',
      component: TaskList
    },
  ],
})
```

โดยค่า `import.meta.env.BASE_URL` จะมีค่าเท่ากับ **base** ที่ระบุใน step 1
(ไม่ต้องกำหนดขึ้นมาเอง)

3. ทดสอบบนเครื่องผู้พัฒนา จะใช้ `npm run dev` ซึ่งจะไม่ได้กำหนดค่า `base` และ `path` ใน `router` ยังกำหนดไว้เหมือนเดิม (ไม่ได้เพิ่มค่า `base` เข้าไปใน `path` เอง จึงใช้ได้ทั้งในขั้นตอนพัฒนาและตอน `deploy` ขึ้น `dev server`)

4. กำหนดค่า URL ของ API ให้แตกต่างกัน ระหว่างตอนทดสอบบนเครื่องผู้พัฒนา (dev mode) กับตอน deploy ขึ้น server (production mode, ใช้ public url)
- สร้างไฟล์ชื่อ `.env` และ `.env.production` ไว้ใน project root โดย `.env` จะถูกเรียกตอน build ใน dev mode และ `.env.production` จะถูกเรียกใน production mode

`.env`

```
VITE_API_ROOT=http://localhost:8080/v1
```

`.env.production`

```
# VITE_API_ROOT=http://ip23ft.sit.kmutt.ac.th:8080/v1  
VITE_API_ROOT=/ft/api/v1
```

`using the variable`

```
const API_ROOT = import.meta.env.VITE_API_ROOT;  
const res = await fetch(API_ROOT+"/tasks");
```

1. ไม่จำเป็นต้องแก้ไข code ในส่วน backend สามารถเรียกใช้ผ่าน public url เช่น <http://intproj23.sit.kmutt.ac.th:8080/<TEAMCODE>/v1/tasks> ได้เลย แต่หากมีการระบุ origin ใน `@CrossOrigin` ให้เพิ่ม public url เข้าไปด้วย

sample 'taskController.java'

```
@CrossOrigin(origins = {"http://localhost:5173",  
                        "http://ip23ft.sit.kmutt.ac.th",  
                        "http://intproj23.sit.kmutt.ac.th"})
```


2. เมื่อ deploy บน VM และระบุค่า **base** ตอน build frontend แล้ว
เมื่อเรียกผ่าน private url เช่น <http://ip23ft.sit.kmutt.ac.th> ระบบจะทำงานไม่ถูกต้อง
เนื่องจาก request จะเข้ามาที่ path **/base/** ซึ่งไม่มี ทำให้ต้องเรียกผ่าน nested public
path เท่านั้น

สามารถแก้ไขด้วย reverse proxy เพื่อให้เมื่อเรียกผ่าน **/base/** บน private url ก็ยังไปที่
frontend เช่นเดียวกัน

3. เมื่อใช้ค่า **base** แล้ว แนะนำให้เรียกแต่ละหน้าด้วย named routes เนื่องจากว่าจะไม่ต้องใส่ค่า **base** ไว้ใน path เอง และหากเราจำเป็นต้องเปลี่ยน path ของแต่ละหน้า ก็จะสามารถเปลี่ยนที่ router ที่เดียว

```
using router.push
```

```
<script setup>
...
import router from '../router'
...
alert("The requested page is not available!")
router.push({name: 'taskList'})
```