

Do it! Vue.js 입문 - 04장

라우터

- 라우터 : 여러 페이지 간에 이동하는 방법
- 싱글 페이지 애플리케이션에서 모두 라우터를 사용

뷰 라우터

- 뷰에서 제공하는 공식 라우팅 라이브러리
- 아래 링크로 라우터를 설치

[뷰 라우터 설치 페이지](#)

뷰 라우터 형식

- router-link 태그는 to에 지정된 URL로 주소를 갱신해줌

```
<router-link to="URL 값">
```

- router-view 태그는 갱신된 URL에 해당하는 컴포넌트를 표시

```
<router-view></router-view>
```

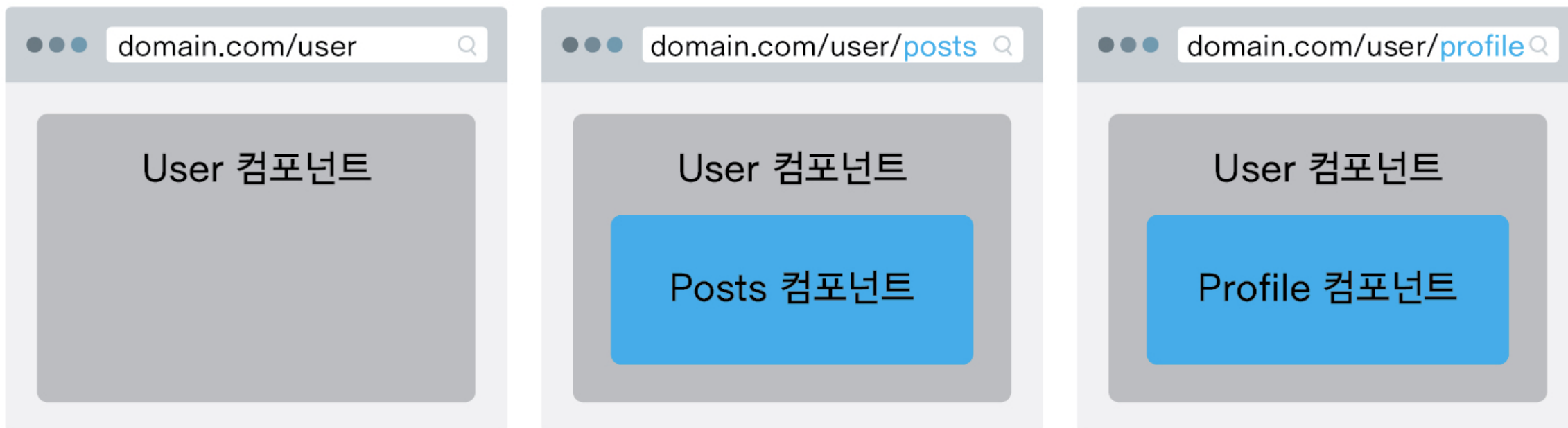
뷰 라우터 예제

- 아래 링크로 뷰 라우터 실습 코드를 실행

[뷰 라우터 샘플 코드](#)

네스티드 라우터

- 네스티드 라우터는 특정 URL에서 하위 컴포넌트를 포함한 상위 컴포넌트를 표시하는 라우팅 방식



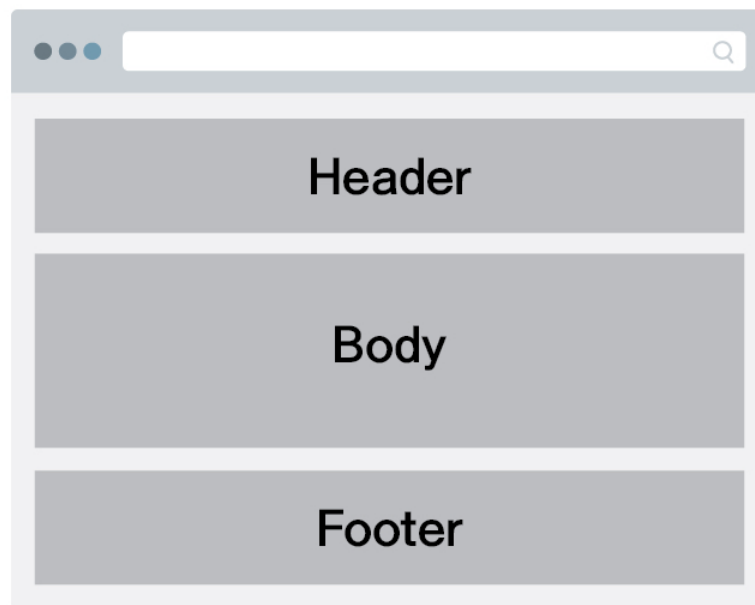
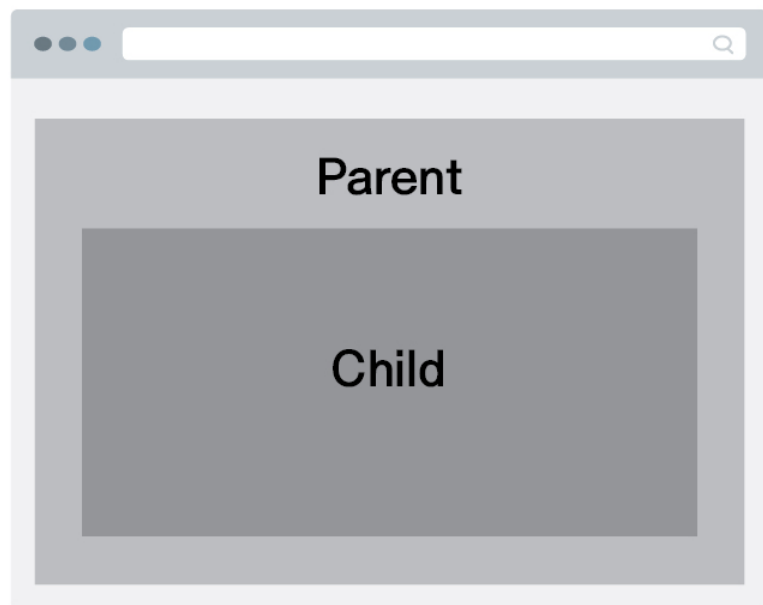
네스티드 라우터 코드 형식

- 상위 컴포넌트는 User 하위 컴포넌트는 UserPost, UserProfile인 경우

```
var routes = [{  
  path: '/user',  
  component: User,  
  children: [  
    { path: 'posts', component: UserPost },  
    { path: 'profile', component: UserProfile }  
  ]  
}];
```

네임드 뷰

- 특정 페이지로 이동했을 때 여러 개의 컴포넌트를 동시에 표시



네스티드 라우터 vs 네임드 뷰

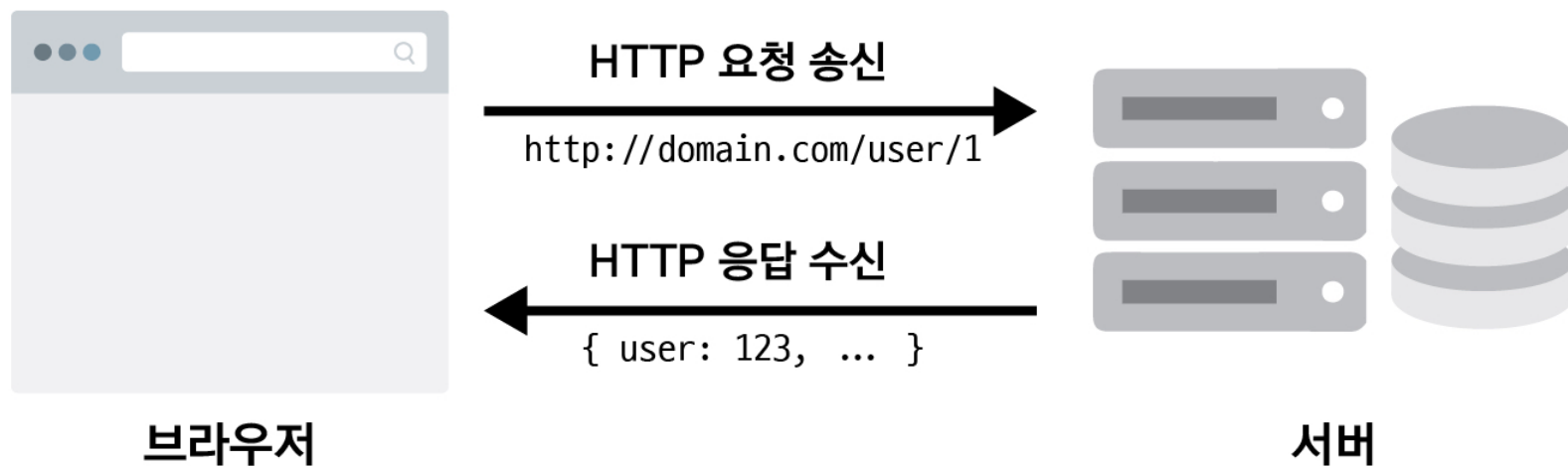
네임드 뷰 코드 형식

```
<router-view name="appHeader"></router-view>
<router-view></router-view> <!-- default에 해당하는 컴포넌트 -->
<router-view name="appFooter"></router-view>
```

```
var routes = [{
  path: '/',
  components: {
    appHeader: Header,
    default: Body,
    appFooter: Footer
  }
}];
```

뷰 HTTP 통신

- 서버에 저장한 데이터를 화면으로 가져오는 방법
- HTTP : 브라우저와 서버 간에 데이터를 주고 받는 통신 프로토콜



뷰 리소스

- 초창기 뷰 공식 HTTP 통신 라이브러리였으나 지금은 제외
- ajax 통신 기반으로 라이브러리가 제작되어 있음
- 아래 링크로 설치

[뷰 리소스 설치 페이지](#)

뷰 리소스 사용 예제

- 라이브러리를 로딩하고 나서 아래와 같이 this.\$http로 접근

```
this.$http.get('url 주소').then(function (response) {  
  console.log(response);  
}).catch(function (err) {  
  console.error(err);  
})
```

액시오스

- 뷰 커뮤니티에서 가장 많이 쓰는 HTTP 통신 라이브러리
- Promise 기반 API를 제공하여 사용성이 뛰어남
- 아래 링크로 설치

[액시오스 설치 페이지](#)

엑시오스 사용 예제

- 라이브러리를 로딩하고 나서 아래와 같이 axios로 접근

```
axios.get('url 주소').then().catch();  
axios.post('url 주소').then().catch();
```

끝