Versioning

Mash-up 11th 노드팀 엄신영

Agenda

1. Versioning Type

2. Versions

Versioning Type

1. URI Versioning Type

2. Header Versioning Type

3. Media Type Versioning Type

1. URI Versioning Type

https://example.com/v1/route

https://example.com/v2/route

```ts
main.ts                                                              JS

const app = await NestFactory.create(AppModule);
// or "app.enableVersioning()"
app.enableVersioning({
  type: VersioningType.URI,
});
await app.listen(3000);
```

2. Header Versioning Type

사용자 지정 요청 헤더를 사용하여,
헤더 값이 요청에 사용할 버전을 지정

```ts
// main.ts                                              JS

const app = await NestFactory.create(AppModule);
app.enableVersioning({
  type: VersioningType.HEADER,
  header: 'Custom-Header',
});
await app.listen(3000);
```

3. Media Type Versioning Type

요청의 Accept 헤더를 사용하여 버전을 지정

```
main.ts                                                    JS

const app = await NestFactory.create(AppModule);

app.enableVersioning({
  type: VersioningType.MEDIA_TYPE,
  key: 'v=',                          ◀──────── Accept: application/json;v=2
});

await app.listen(3000);
```

Controller versions

```ts
cats.controller.ts                                          JS

@Controller({
  version: '1',
})
export class CatsControllerV1 {
  @Get('cats')
  findAll(): string {
    return 'This action returns all cats for version 1';
  }
}
```

Route versions

```ts
cats.controller.ts                                    JS


import { Controller, Get, Version } from '@nestjs/common';


@Controller()
export class CatsController {
  @Version('1')
  @Get('cats')
  findAllV1(): string {
    return 'This action returns all cats for version 1';
  }


  @Version('2')
  @Get('cats')
  findAllV2(): string {
    return 'This action returns all cats for version 2';
  }
}
```

Multiple versions

```ts
cats.controller.ts                                    JS

@Controller({
  version: ['1', '2'],
})
export class CatsController {
  @Get('cats')
  findAll(): string {
    return 'This action returns all cats for version 1 or 2';
  }
}
```

Version "Neutral"

```js
cats.controller.ts                                                    JS

import { Controller, Get, VERSION_NEUTRAL } from '@nestjs/common';

@Controller({
  version: VERSION_NEUTRAL,
})
export class CatsController {
  @Get('cats')
  findAll(): string {
    return 'This action returns all cats regardless of version';
  }
}
```

감사합니다.