عنوان: Best Practice هایی برای طراحی RESTful API - قسمت دوم
نویسنده: محسن درپرستی
تاریخ: ۱۳۹۲/۱۲/۰۳ ۱۴:۰
آدرس: www.dotnettips.info
گروه‌ها: ASP.NET Web API, Naming, Web API, RESTful API, Best Practice

طراحی Url در Restful API

Url بخش اصلی و راه ارتباطی API شما با توسعه دهنده است .بنابراین طراحی یک ساختار مناسب و یکپارچه برای Url ها دارای اهمیت زیادی است .

Url پایه API خود را ساده و خوانا ، حفظ کنید . داشتن یک Url پایه ساده استفاده از API را آسان کرده و خوانایی آن را بالا میبرد و باعث می‌شود که توسعه دهنده برای استفاده از آن نیاز کمتری به مراجعه به مستندات داشته باشد. پیشنهاد می‌شود که برای هر منبع تنها دو Url پایه وجود داشته باشد . یکی برای مجموعه ای از منبع موردنظر و دیگری برای یک واحد مشخص از آن منبع .
برای مثال اگر منبع موردنظر ما کتاب باشد ، خواهیم داشت :

```
.../books
```

برای مجموعه‌ی کتابها و

```
.../books/1001
```

برای کتابی با شناسه 1001
استفاده از این روش یک مزیت دیگر هم به همراه دارد و آن دور کردن افعال از Url ها است.

بسیاری در زمان طراحی Url ها و در نامگذاری از فعل‌ها استفاده می‌کنند. برای هر منبعی که مدلسازی می‌کنید هیچ وقت نمی‌توانید آن را به تنهایی و جداافتاده در نظر بگیرید. بلکه همیشه منابع مرتبطی وجود دارند که باید در نظر گرفته شوند. در مثال کتاب می‌توان منابعی مثل نویسنده ، ناشر ، موضوع و ... را بیان کرد. حالا سعی کنید به تمام Url هایی که برای پوشش دادن تمام درخواست‌های مربوط به منبع کتاب نیاز داریم فکر کنید . احتمالا به چیزی شبیه این می‌رسیم :

```
.../getAllBooks
.../getBook
.../newBook
.../getNewBooksSince
.../getComputerBooks
.../BooksNotPublished
.../UpdateBookPriceTo
.../bookForPublisher
.../GetLastBooks
.../DeleteBook
….
```

خیلی زود یک لیست طولانی از Url ها خواهید داشت که به علت نداشتن یک الگوی ثابت و مشخص استفاده از API شما را واقعا سخت می‌کند.

پس حالا این درخواست‌های متنوع را چطور با دو Url اصلی انجام دهیم ؟
-1 از افعال Http برای کار کردن بر روی منابع استفاده کنید . با استفاده از افعال Http شامل PUT ، GET ، POST و DELETE و دو Url اصلی ، یک مجموعه‌ی مناسب از عملیات‌ها در دسترس توسعه دهنده خواهد بود . به جدول زیر نگاه کنید .

منبع	POST Create	GET Read	PUT Update	DELETE Delete
/books	ثبت کتاب جدید	لیست کتابها	بروزرسانی کلی کتابها	حذف تمام کتابها
/books/1001	خطا	نمایش کتاب ۱۰۰۱	اگر وجود داشته باشد بروزرسانی وگرنه خطا	حذف کتاب ۱۰۰۱

توسعه دهندگان احتمالا نیازی به این جدول برای درک اینکه API چطور کار می‌کند نخواهند داشت.

-2 با استفاده از نکته قبلی بخشی از Url های بالا حذف خواهند شد. اما هنوز با روابط بین منابع چکار کنیم؟ منابع تقریبا همیشه دارای روابطی با دیگر منابع هستند . یک روش ساده برای بیان این روابط در API چیست ؟ به مثال کتاب برمیگردیم. کتاب‌ها دارای نویسنده هستند. اگر بخواهیم کتاب‌های یک نویسنده را برگردانیم چه باید بکنیم؟ با استفاده از Url های پایه و افعال Http می‌توان اینکار را انجام داد. یکی از ساختارهای ممکن این است :

```
GET .../authors/1001/books
```

اگر بخواهیم یک کتاب جدید به کتابهای این نویسنده اضافه کنیم :

```
POST .../authors/1001/books
```

و حدس زدن اینکه برای حذف کتابهای این نویسنده چه باید کرد ، سخت نیست .

-3 بیشتر API ها دارای پیچیدگی‌های بیشتری نسبت به Url اصلی یک منبع هستند . هر منبع مشخصات و روابط متنوعی دارد که قابل جستجو کردن، مرتب سازی، بروزرسانی و تغییر هستند. Url اصلی را ساده نگه دارید و این پیچیدگی‌ها را به کوئری استرینگ منتقل کنید.

برای برگرداندن تمام کتابهای با قیمت پنج هزار تومان با قطع جیبی که دارای امتیاز 8 به بالا هستند از کوئری زیر می‌شود استفاده کرد :

```
GET .../books?price=5000&size=pocket&score=8
```

و البته فراموش نکنید که لیستی از فیلدهای مجاز را در مستندات خود ارائه کنید.

4 - گفتیم که بهتر است افعال را از Url ها خارج کنیم . ولی در مواردی که درخواست ارسال شده در مورد یک منبع نیست چطور؟ مواردی مثل محاسبه مالیات پرداختی یا هزینه بیمه ، جستجو در کل منابع ، ترجمه یک عبارت یا تبدیل واحدها . هیچکدام از اینها ارتباطی با یک منبع خاص ندارند. در این موارد بهتر است از افعال استفاده شود. و حتما در مستندات خود ذکر کنید که در این موارد از افعال استفاده می‌شود.

```
.../convert?value=25&from=px&to=em
.../translate?term=web&from=en&to=fa
```

5 - استفاده از اسامی جمع یا مفرد

با توجه به ساختاری که تا اینجا طراحی کرده ایم بکاربردن اسامی جمع بامعناتر و خواناتر است. اما مهمتر از روشی که بکار می‌برید ، اجتناب از بکاربردن هر دو روش با هم است ، اینکه در مورد یک منبع از اسم منفرد و در مورد دیگری از اسم جمع استفاده کنید . یکدستی API را حفظ کنید و به توسعه دهنده کمک کنید راحت‌تر API شما را یاد بگیرد.

-6 استفاده از نام‌های عینی به جای نام‌های کلی و انتزاعی

API ی را در نظر بگیرید که محتواهایی را در فرمت‌های مختلف ارائه می‌دهد. بلاگ ، ویدئو ، اخبار و .... حالا فرض کنیداین API منابع را در بالاتری سطح مدسازی کرده باشد مثل /items یا /assets . درک کردن محتوای این API و کاری که می‌توان با این API انجام داد برای توسعه دهنده سخت است . خیلی راحت‌تر و مفیدتر است که منابع را در قالب بلاگ ، اخبار ، ویدئو مدلسازی کنیم .