

## Pipes

There is no fundamental difference between [regular pipes](#) and microservices pipes. The only difference is that instead of throwing `HttpException`, you should use `RpcException`.

**info Hint** The `RpcException` class is exposed from `@nestjs/microservices` package.

### Binding pipes

The following example uses a manually instantiated method-scoped pipe. Just as with HTTP based applications, you can also use controller-scoped pipes (i.e., prefix the controller class with a `@UsePipes()` decorator).

```
@@filename()
@UsePipes(new ValidationPipe())
@MessagePattern({ cmd: 'sum' })
accumulate(data: number[]): number {
  return (data || []).reduce((a, b) => a + b);
}

@@switch
@UsePipes(new ValidationPipe())
@MessagePattern({ cmd: 'sum' })
accumulate(data) {
  return (data || []).reduce((a, b) => a + b);
}
```