Models in ASP.NET Web API

Shawn Wildermuth

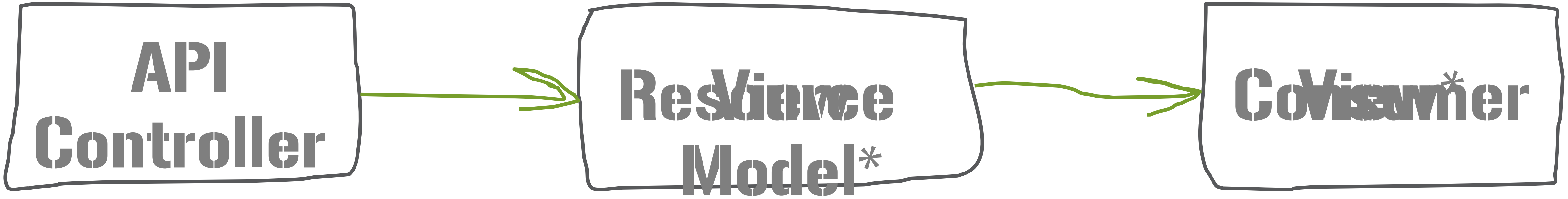
@shawnwildermuth | wilderminds.com

Module Overview

Models in ASP.NET Web API

- The Role of Models in Web API
- Models and Resources
- Validation in Web API
- Versioning of Models

Models in Web API



API Controller

Resource

View Model*

Consumer

View*

What Do We Call Them?

- View Model
  - *Data that supplied to a view*
- Binding Model
  - *Data that is supplied for data binding*
- Resource Model
  - *Data that is supplied through an API*

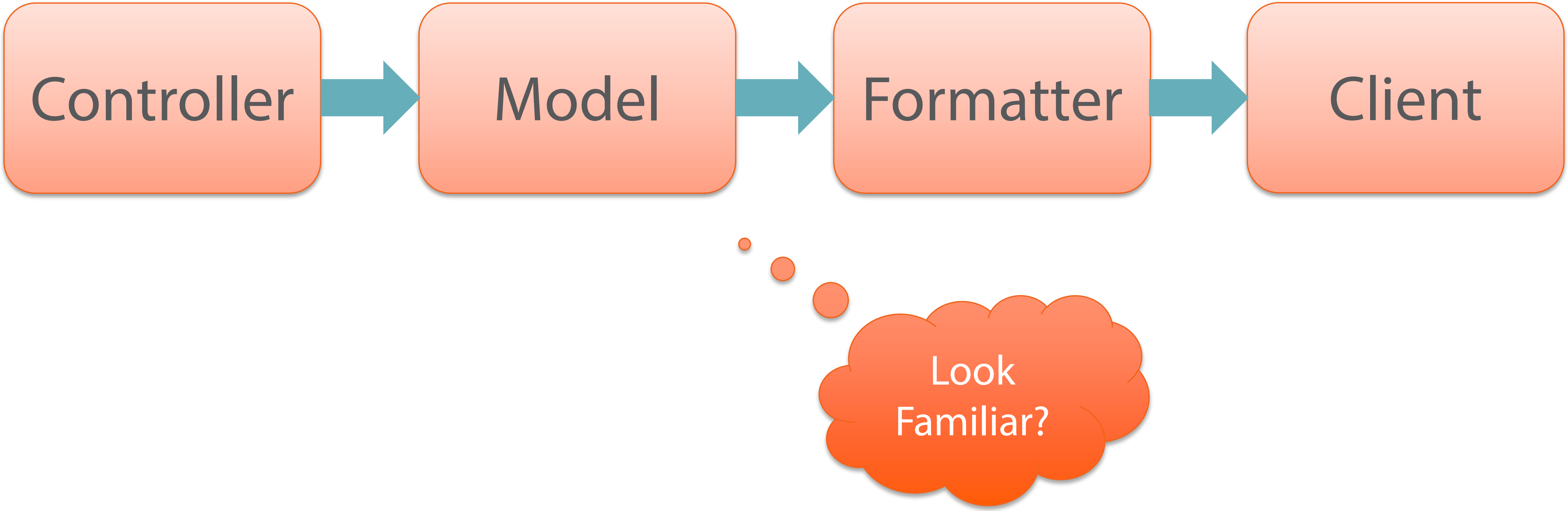
WHO CARES?

Don’t get caught up in the dogma.

The requirement to send and receive data means models will help you do your job.

Period.

Models in Web API



Controller → Model → Formatter → Client

Look Familiar?

Design More Than Your API

```
/api/profile

/api/profile/interests

/api/profile/photos

/api/profile/photos/1
```

Design More Than Your API

```
{
  "pitch": "This is a new pitch",
  "lookingFor": "A woman, any woman",
  "introduction": "Family is very important.",
  "birthdate": "1990-05-17T00:00:00",
  "orientation": "Straight",
  "gender": "Male",
  "memberName": "awkwardguy",
  "interests": [ ... ],
  "photos": [ ... ]
}
```

Demo

Using Models in Your API

Demo

Validation on an API

Demo

Collections of View Models

Unlike View Models in MVC,
your API Models are a contract.
You can’t break them!

Model Versioning

- It’s too easy to just version your API
  - You must version your models too
  - Typically done with media type headers

```
HTTP/1.1 200 OK
Content-Type: application/vnd.dmp-com.profile+json; version=1.0
{ "memberName": "awkwardguy", ... }
```

For a complete how-to on
versioning models,
see module 5 of my
Implementing Web API Course:

http://shawnw.me/**pswebapi**

What We've Learned

Models in ASP.NET Web API

The nomenclature doesn't matter; be pragmatic

Separating the Entities from the Contract is beneficial

Validation is still a requirement

You should version your models too