

# Introduction to .NET

Florin Olariu

“Alexandru Ioan Cuza”, University of Iași

Department of Computer Science

# ASP.NET Web API

# Agenda

- ▶ Demo : create an empty API project
- ▶ Using middleware for building an API
- ▶ Demystify MVC pattern
- ▶ Learning about routing
- ▶ Demo : Building and Designing API's
- ▶ The importance of status codes
- ▶ Demo : Returning correct status codes
- ▶ Summary

Demo : create an empty API project

# Using middleware for building an API

The background of the slide features abstract, overlapping geometric shapes in various shades of blue, ranging from light sky blue to deep navy blue. These shapes are primarily located on the right side and bottom of the slide, creating a modern, layered effect.

# Using middleware for building an API

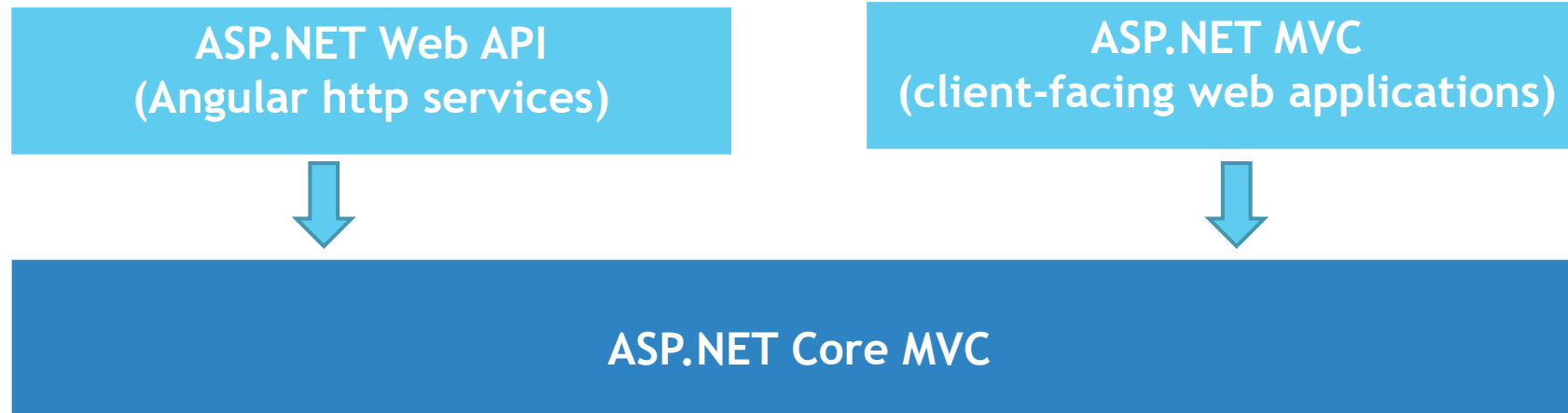
ASP.NET Web API  
(Angular http services)

# Using middleware for building an API

ASP.NET Web API  
(Angular http services)

ASP.NET MVC  
(client-facing web applications)

# Using middleware for building an API





# Demystify MVC pattern

The background of the slide features abstract, overlapping geometric shapes in various shades of blue, ranging from light sky blue to deep navy blue. These shapes are primarily located on the right side and bottom of the frame, creating a modern, dynamic aesthetic.

# Demystify MVC pattern

- ▶ Model-View-Controller

# Demystify MVC pattern

- ▶ Model-View-Controller
- ▶ Architectural pattern

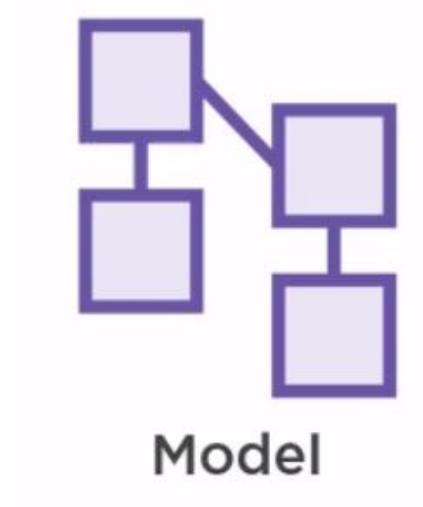
# Demystify MVC pattern

- ▶ Model-View-Controller
- ▶ Architectural pattern
- ▶ Used for: low coupling, separation of concerns

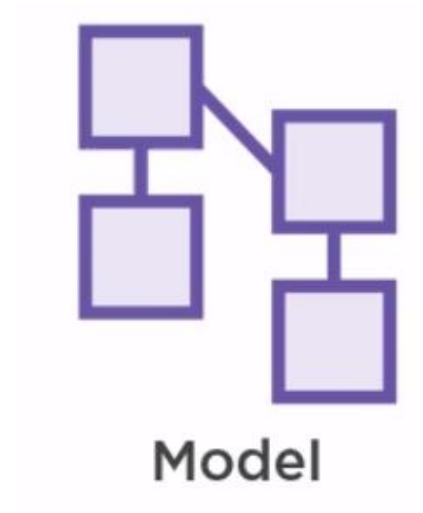
# Demystify MVC pattern

- ▶ Model-View-Controller
- ▶ Architectural pattern
- ▶ Used for: low coupling, separation of concerns
- ▶ It is not a FULL APPLICATION ARCHITECTURE.

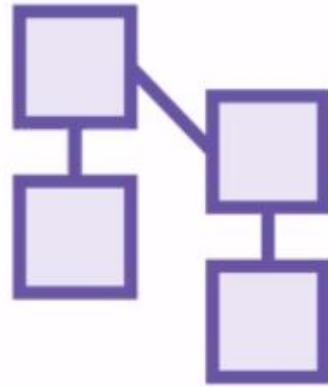
# Demystify MVC pattern



# Demystify MVC pattern



# Demystify MVC pattern



Model



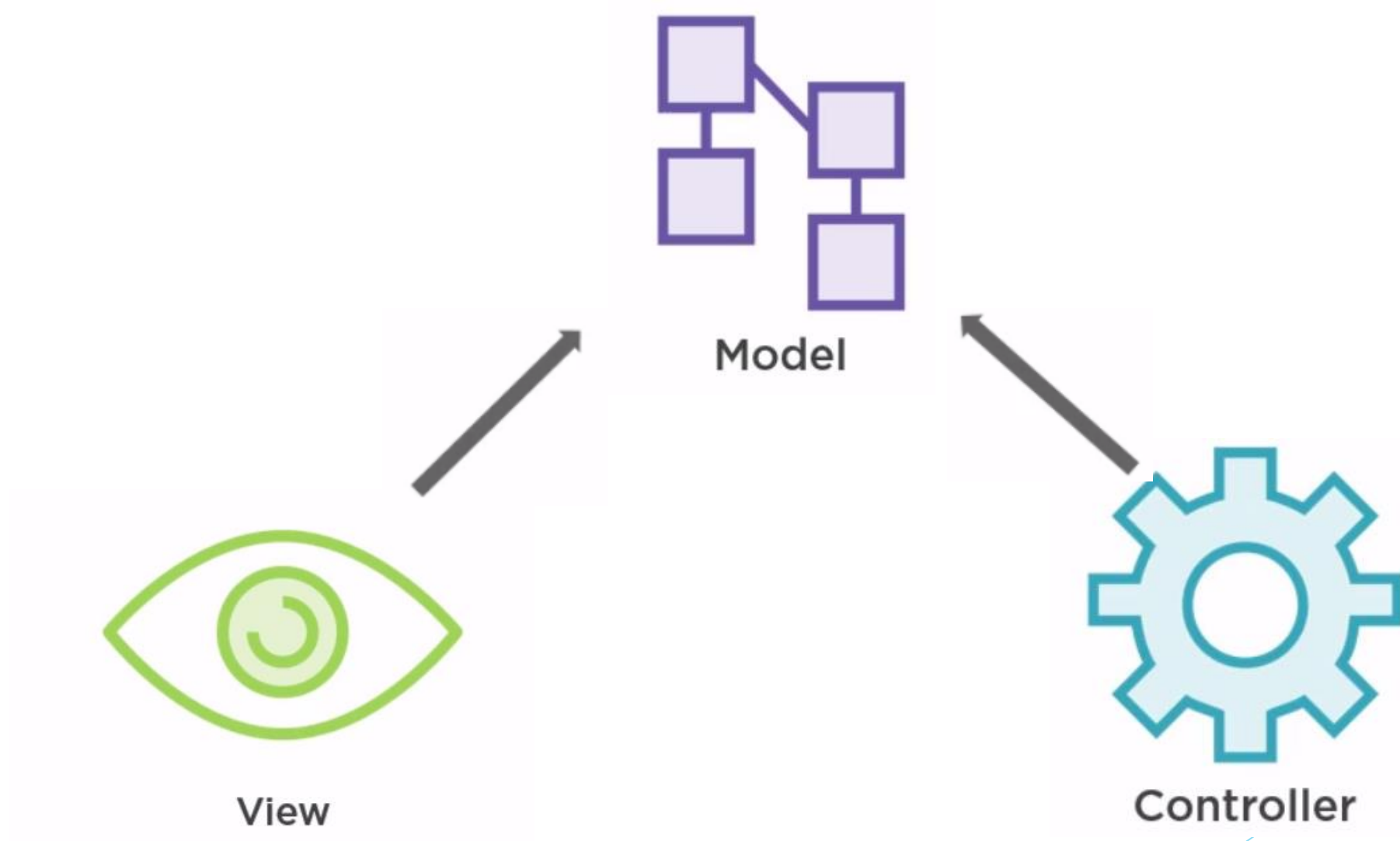
View



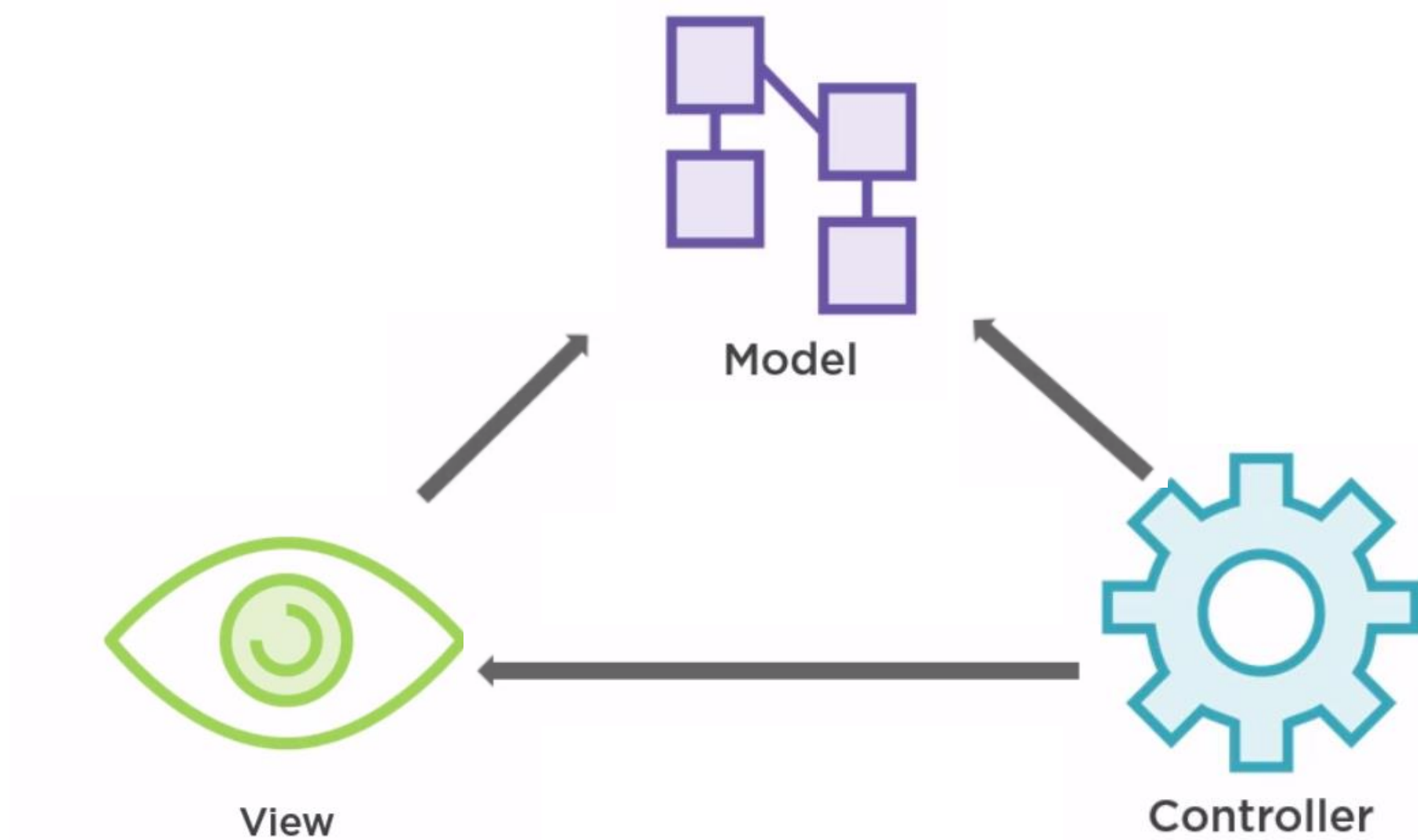
Controller



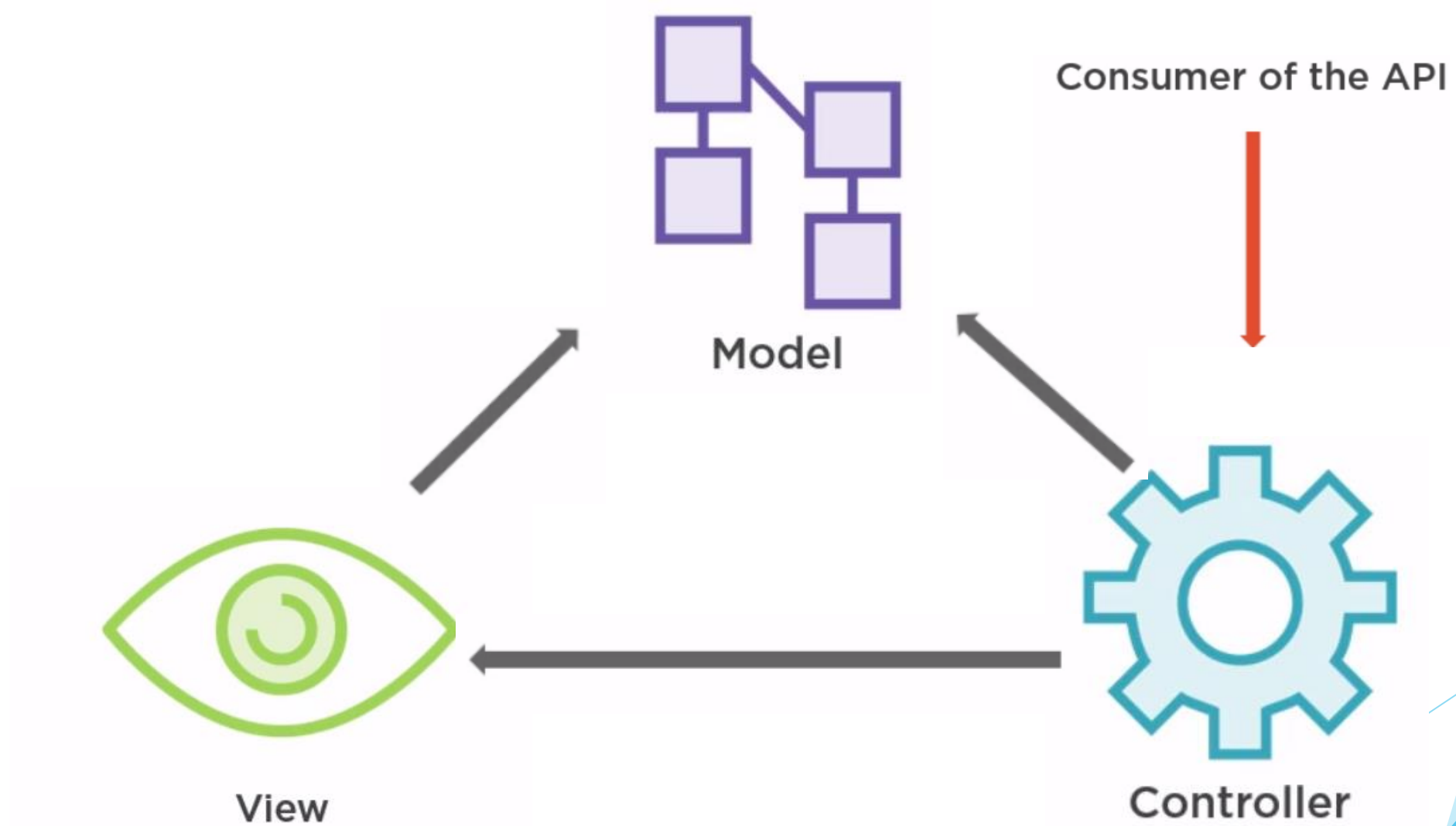
# Demystify MVC pattern



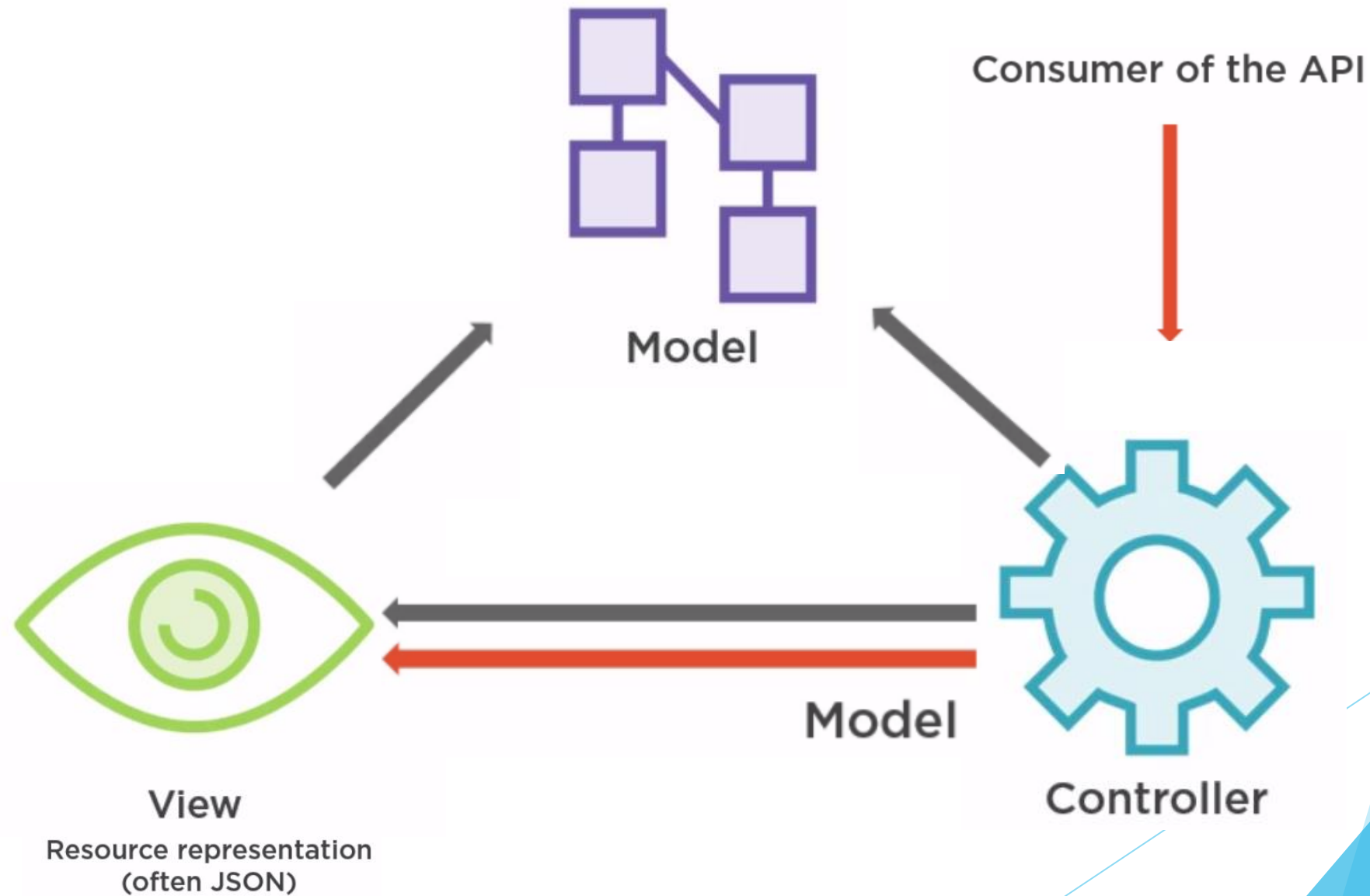
# Demystify MVC pattern



# Demystify MVC pattern



# Demystify MVC pattern



# Demo: Adding ASP.NET Core MVC middleware

# Learning about routing



# Learning about routing

- ▶ Matches request URI to controller method

# Learning about routing

- ▶ Matches request URI to controller method
- ▶ Convention based and attribute based routing



# Learning about routing

- ▶ Matches request URI to controller method
- ▶ Convention based and attribute based routing
  - ▶ Convention based

```
app.UseMvc(routes =>
{
    routes.MapRoute(
        name: "default",
        template: "{controller=Home}/{action=Index}/{id?}");
});
```

# Learning about routing

- ▶ Matches request URI to controller method
- ▶ Convention based and attribute based routing
  - ▶ Attribute based
    - ▶ Attributes at controllers/actions level

# Learning about routing

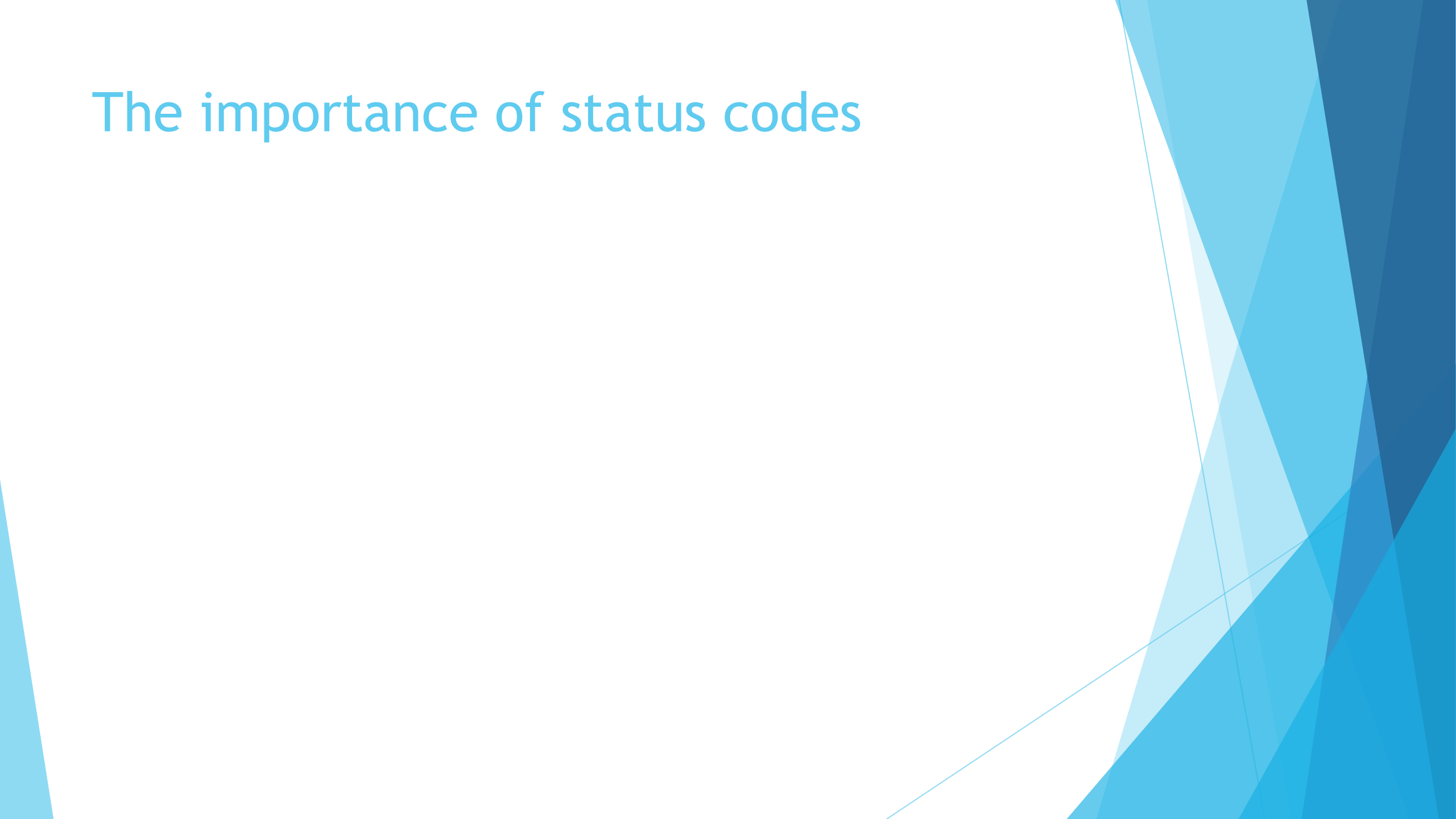
## Routing

HTTP Method	Attribute	Level	Sample URI
GET	HttpGet	Action	/api/cities /api/cities/1
POST	HttpPost	Action	/api/cities
PUT	HttpPut	Action	/api/cities/1
PATCH	HttpPatch	Action	/api/cities/1
DELETE	HttpDelete	Action	/api/cities/1
---	Route	Controller	---

# Demo : Building and Designing API's

- ▶ Improving the architecture with model classes

# The importance of status codes



# The importance of status codes

- ▶ Is a part of a response

# The importance of status codes

- ▶ Is a part of a response
- ▶ Provides information on:

# The importance of status codes

- ▶ Is a part of a response
- ▶ Provides information on:
  - ▶ Whether or not the response worked as expected



# The importance of status codes

- ▶ Is a part of a response
- ▶ Provides information on:
  - ▶ Whether or not the response worked as expected
  - ▶ Who is responsible for a failed request (consumer or our API)

# The importance of status codes

## Level 200 Success

200 – OK  
201 – Created  
204 – No Content

## Level 400 Client Error

400 – Bad Request  
401 – Unauthorized  
403 – Forbidden  
404 – Not Found  
409 – Conflict

## Level 500 Server Error

500 – Internal  
Server Error

# Demo : Returning correct status codes

One more thing...(1/2)

One more thing...(2/2)

Postel's Law:

*“Be conservative in what you do,  
be liberal in what you accept  
from others.”*

# Summary

- ▶ Status codes
- ▶ MVC
- ▶ How to build a proper API
- ▶ Routing

# Bibliography

► Pluralsight

# Questions

- ▶ Do you have any other questions?



Thanks!

See you next time! 😊