ASP.net WEP API

Contents

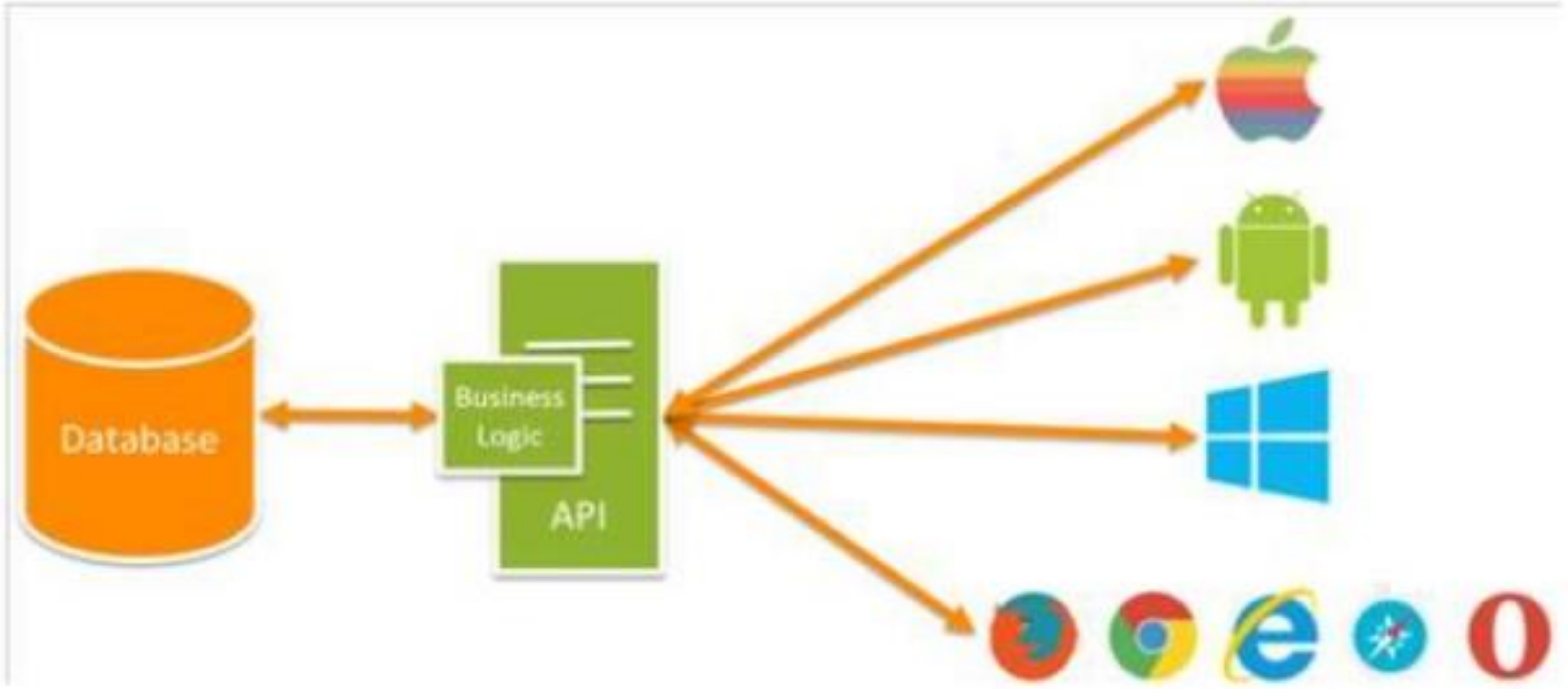
- What is an API?
- Why ASP.NET MVC Web API?
- HyperText Transfer Protocol
- REST
- JSON
- Introduction to Web API

What is an API?

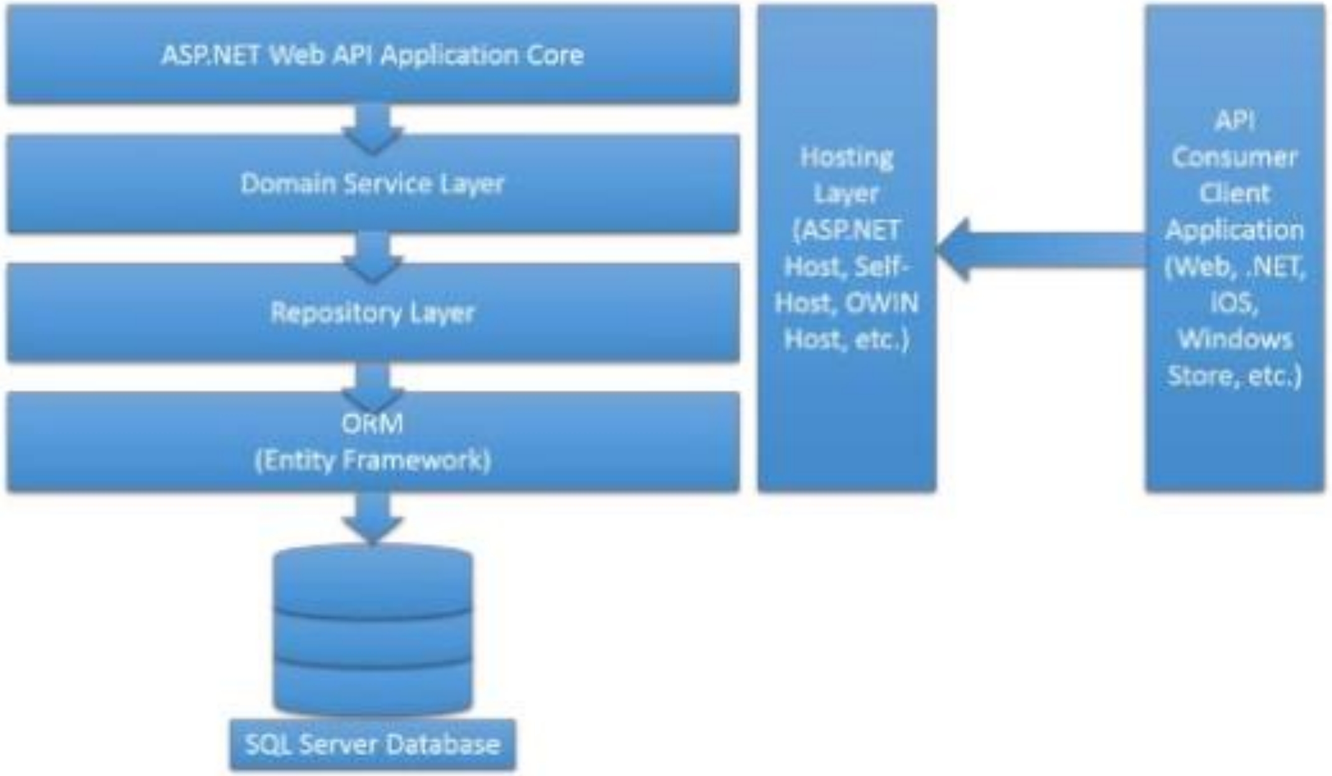
- An **application programming interface** (**API**) is a specification intended to be used as an interface by software components to communicate with each other. An API may include specifications for routines, data structures, object classes, and variables.
- For the web this mean Web Services (SOAP + XML + WSDL) but in Web 2.0 we are moving away to REST Services (HTTP + XML/JSON) – **Web API**
- Web APIs allow the combination of multiple services into new applications known as mashups.

Why ASP.NET MVC Web Api?

- Web Api
  - Defined in HTTP
  - Messages in Json/XML
  - RESTful
  - CRUD operations
  - Ready for Cloud


Database
Business Logic
API

Web API


ASP.NET Web API Application Core
Domain Service Layer
Repository Layer
ORM
(Entity Framework)
SQL Server Database
Hosting Layer (ASP.NET Host, Self-Host, OWIN Host, etc.)
API Consumer Client Application (Web, .NET, iOS, Windows Store, etc.)

Rest - **Re**presentational **S**tate **T**ransfer

- **REST** is a style of software architecture for distributed systems such as the WWW, where, virtually in all cases, the HTTP protocol is used.
- Uses CRUD actions (HTTP methods)

CRUD Action	HTTP Method
Create	Post
Read	Get
Update	Put
Delete	Delete

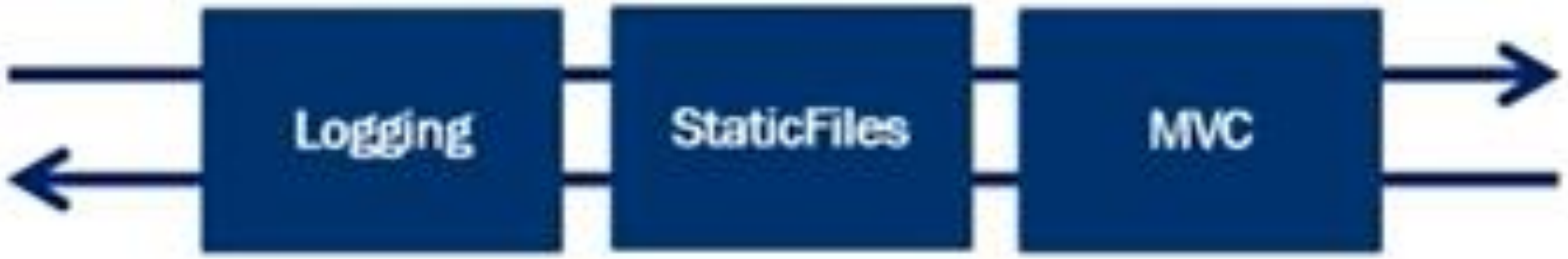
- Each resource is represented by an global id (URI in HTTP)
- The resources are conceptually separate from the representations that are returned to the client (JSON/XML)

In many ways, the World Wide Web itself, based on HTTP, can be viewed as a REST-based architecture.

Despite being simple, REST is fully-featured; there's basically nothing you can do in Web Services that can't be done with a RESTful architecture!

The four main HTTP methods are mapped to CRUD operations:

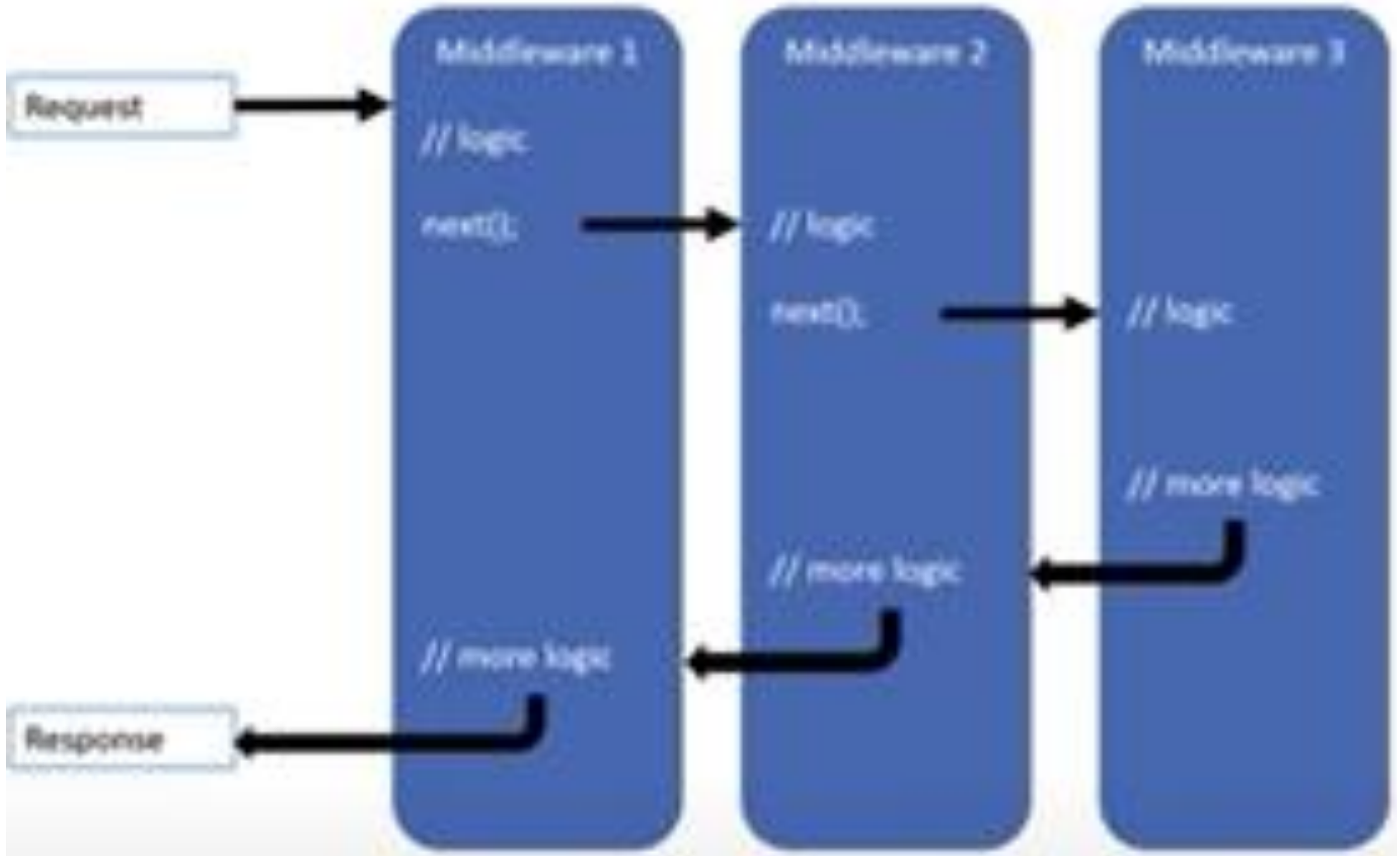
- GET retrieves the representation of the resource at a specified URI. GET should have no side effects on the server.
- PUT updates a resource at a specified URI (idempotent).
- POST creates a new resource. The server assigns the URI for the new object and returns this URI as part of the response message.
- DELETE deletes a resource at a specified URI (idempotent).



Middleware in ASP.NET Core

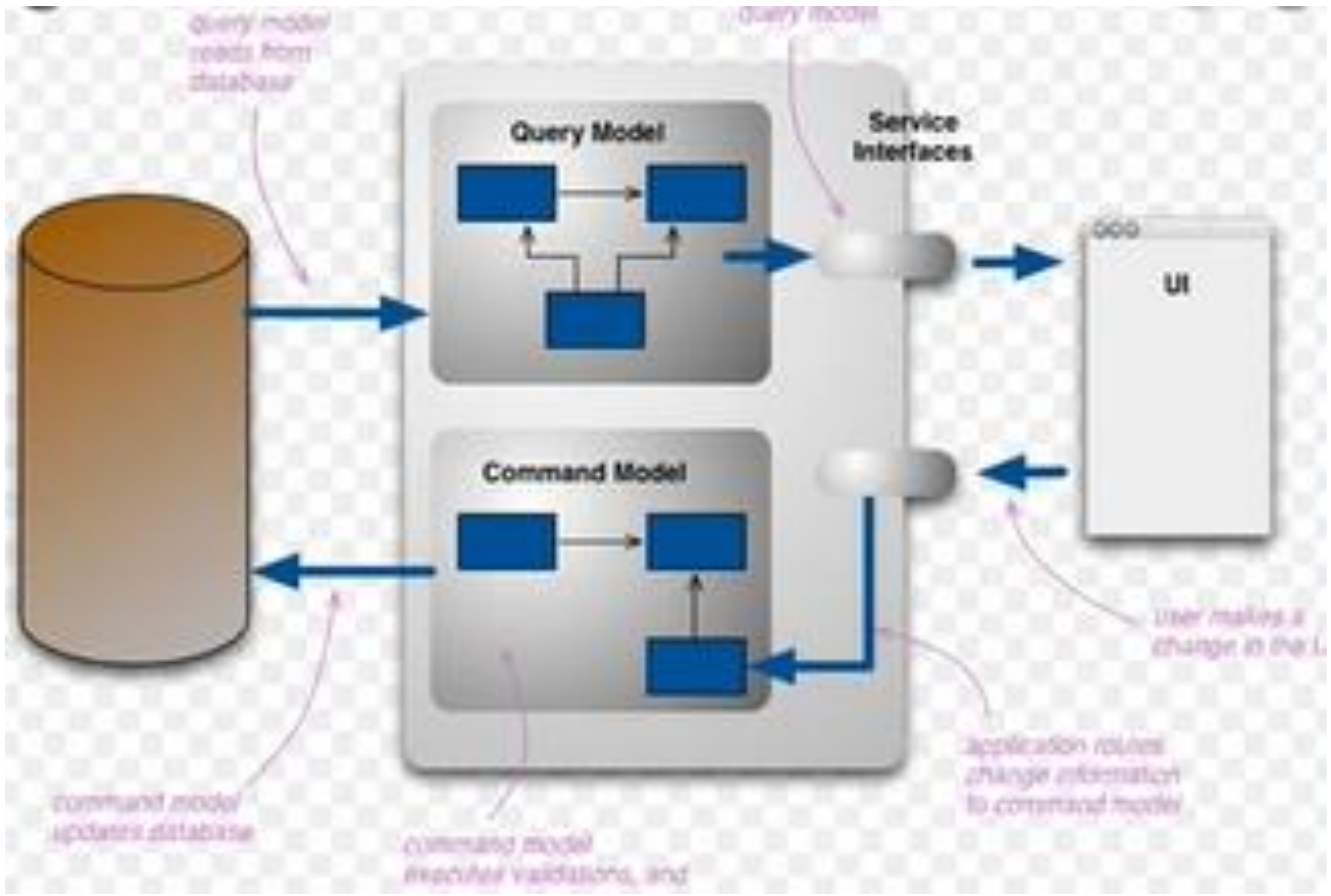
- Has access to both Request and Response
- May simply pass the Request to next Middleware
- May process and then pass the Request to next Middleware
- May handle the Request and short-circuit the pipeline
- May process the outgoing Response
- Middlewares are executed in the order they are added

Configure Request Processing Pipeline



Information Logged

- MW1: Incoming Request
- MW2: Incoming Request
- MW3: Request handled and response produced
- MW2: Outgoing Response


query model
reads from
database
query model
Query Model
Service
Interfaces
UI
Command Model
user makes a
change in the
command model
updates database
command model
executes validations, and
application routes
change information
to command model