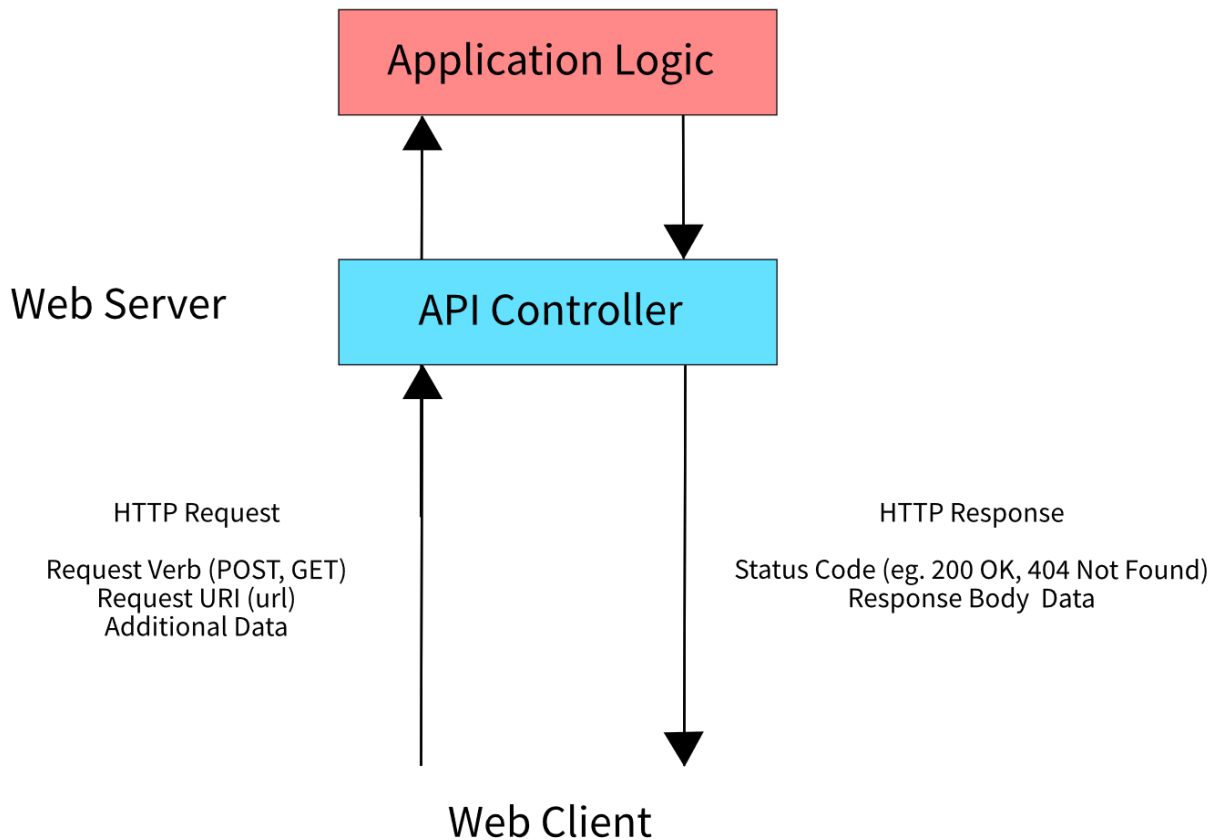


Controllers in ASP.NET

There are two primary types of Controllers in ASP.NET. In the beginning of the semester, we looked at API Controllers, which serve an output given an input.

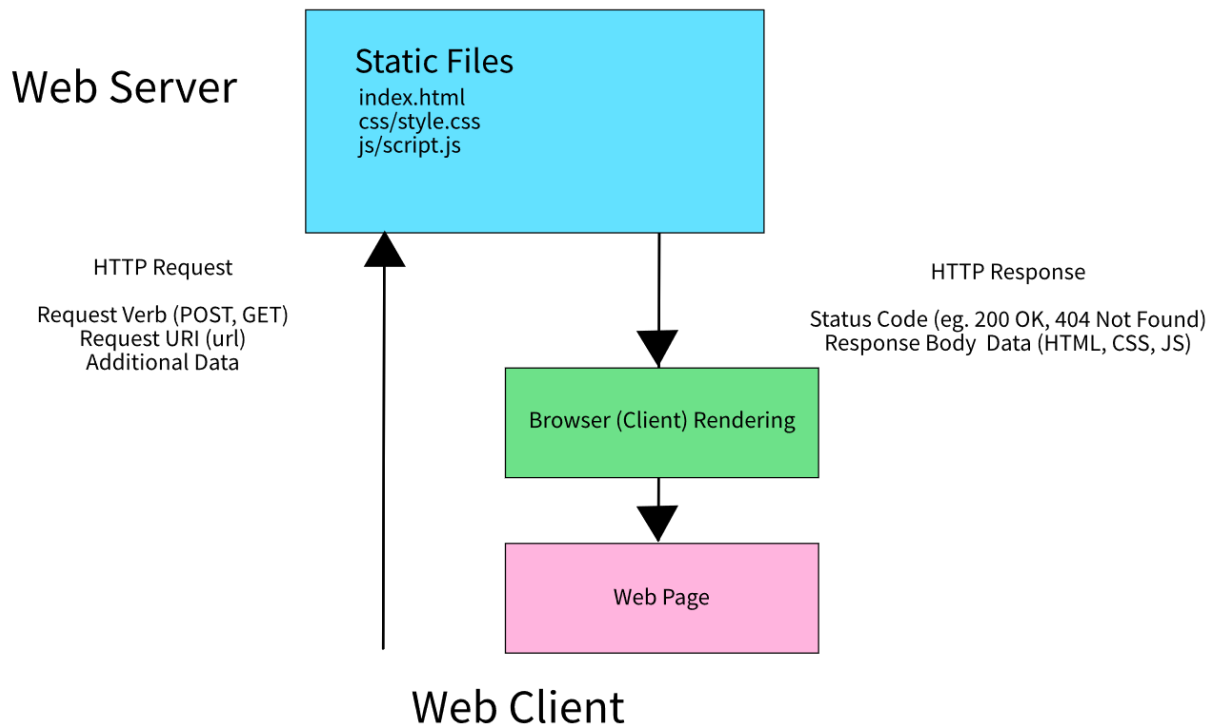


Controllers are used to listen for particular HTTP requests and provide a response.

| | |
|----------|--|
| Request | GET localhost/api/greeting/class/5101 |
| Response | "Hello Class of 5101 - Web Application Development!" |

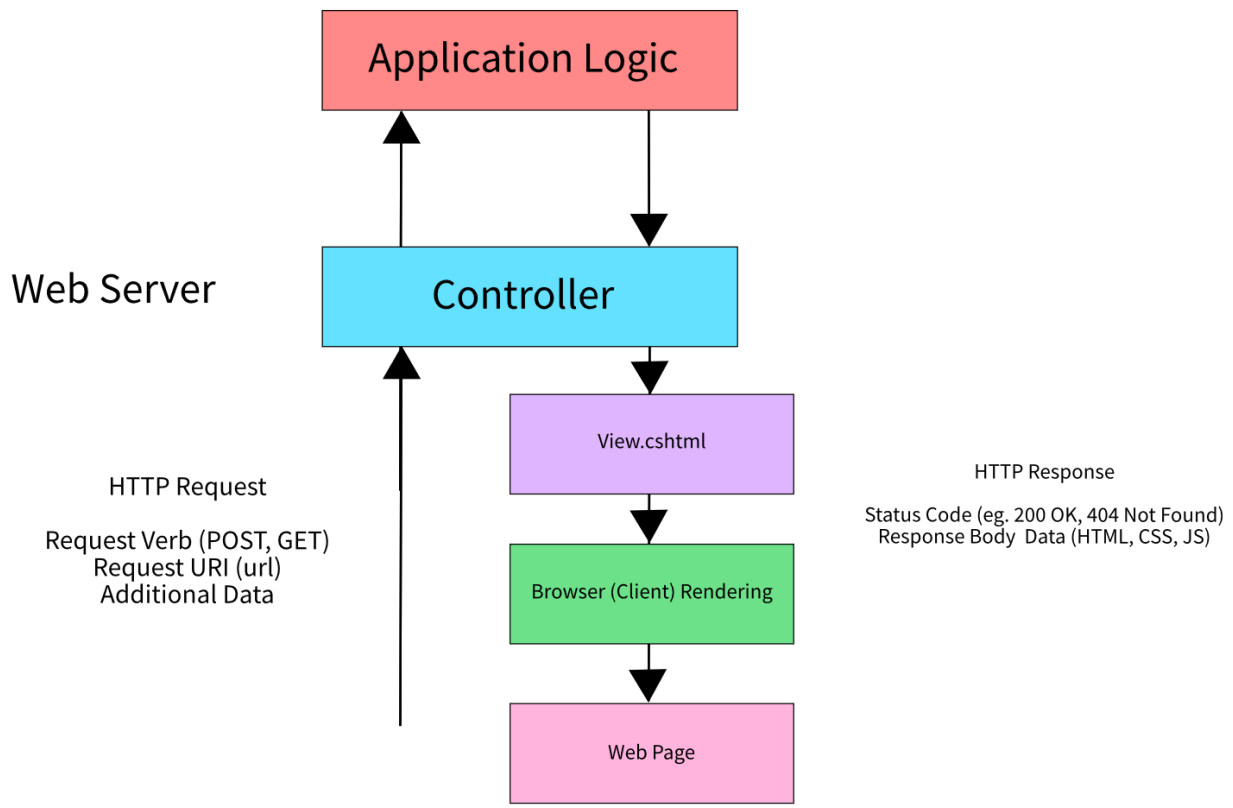
For a WebAPI Controller, the response is standalone data. WebAPI Controllers are designed to serve data. This configuration is known as a '**RE**presentational **S**tate **T**ransfer' (REST-ful) service.

However, recall that web servers have the ability to serve static content such as HTML files, JavaScript Files, and CSS files.



The issue with static websites is that we have no opportunity to introduce any application logic into the web page. With static sites, we can only ask for files which are stored on the web server.

What if we wanted to serve a web page which changes depending on the class code? This web page would change dynamically to serve different messages depending on the input course code. This would be referred to as a dynamic web page.



Web Client

A regular Controller serves the purpose of receiving HTTP requests and returning dynamically rendered content. The Controller will provide necessary information to the View, and the View will contain server instructions that build out an HTML webpage. Once the HTML webpage is built, the response is sent to the client.

| Task | API Controller | Controller |
|------------------------------------|-----------------------------------|-----------------------------|
| Listens for HTTP Request | Yes | Yes |
| Provides HTTP Response | Yes | Yes |
| Executes server logic | Yes | Yes |
| Send instructions to render a page | No | Yes |
| RESTful Service | Yes | No |
| Method Return Type | Data (string, int, bool) | ActionResult |
| Works with Views | No | Yes |
| Customized Routing | App_Start/WebApiConfig.cs | App_Start/RouteConfig.cs |
| Base Class | ControllerName : ApiController | ControllerName : Controller |