

Improving Application Responses with Custom Action Results



Alex Wolf

www.alexwolfthoughts.com

Customer Conversion Rates

Paladin We Fight For You. Always.

[About](#) [Services](#) [Quote](#) [Contact](#)

Thank You! We're so glad you share our vision.

Your estimated rate is **\$102** per month.

Below is a summary of the features you selected. You can adjust this to better fit your needs by tweaking your product selection.

Plan Feature	Cost
Standard Deductible	\$500
Max Coverage	\$100/\$300k
Roadside Assitance	Yes
Car Rental	Yes
Medical Coverage	\$250/500k
New Car Guarantee	Yes

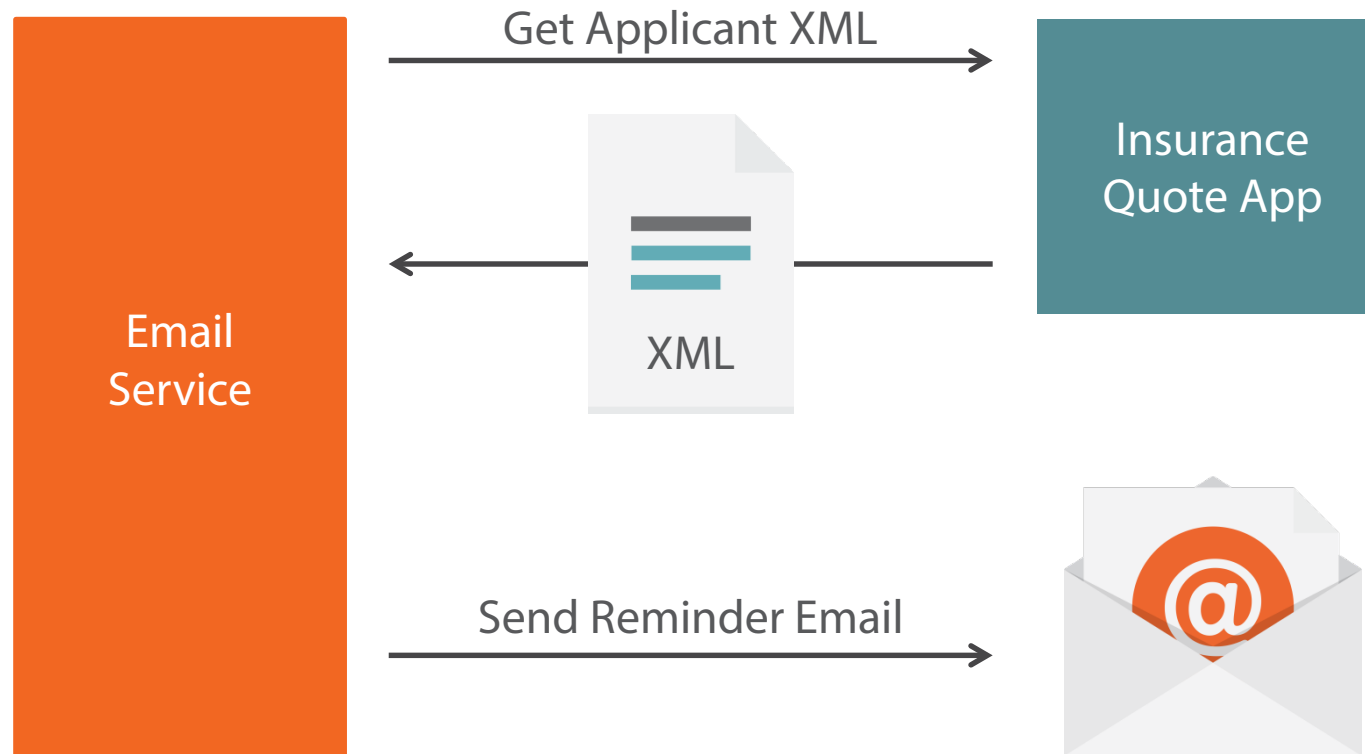
[Edit This Plan](#)

[Sign Me Up!](#)

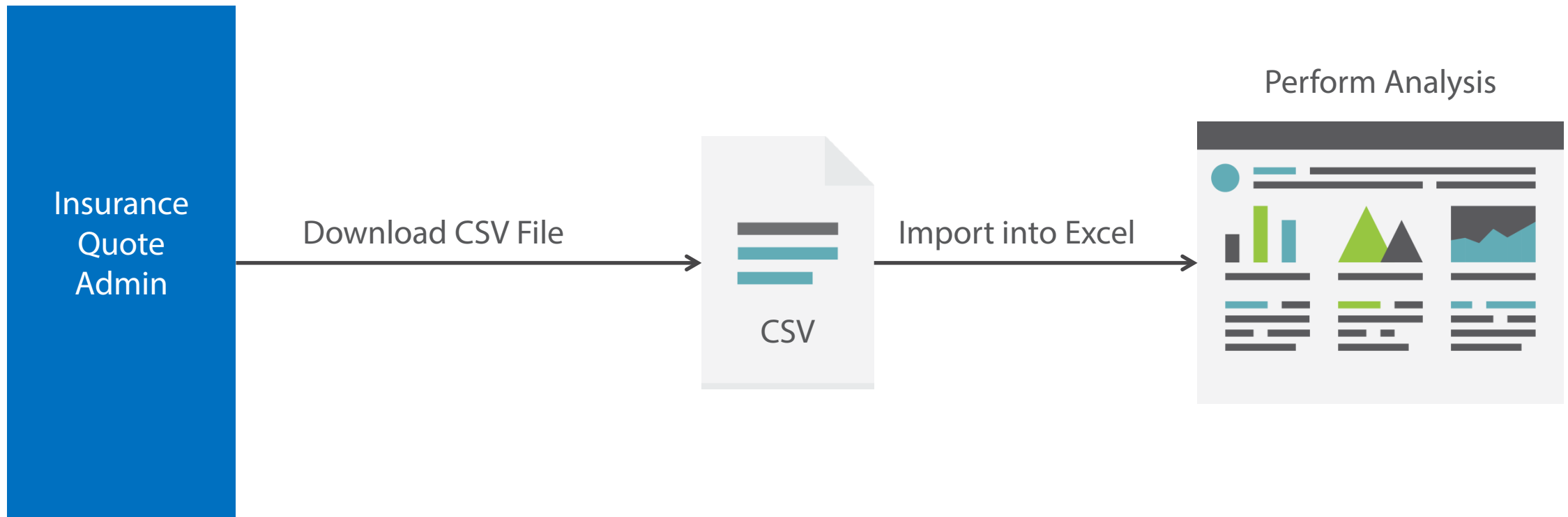


Please Note: Your loan application session has now expired. To update details of this application, contact one of our sales reps. You can also start the process over at any time. Whatever you need - we're there to help you out!

Providing Customer Data



Analyzing Customer Data





Understanding Action Results

Working with Action Results

`return View()`

`==`

`return new ViewResult()`

`return RedirectToAction()`

`==`

`return new
RedirectToRouteResult()`

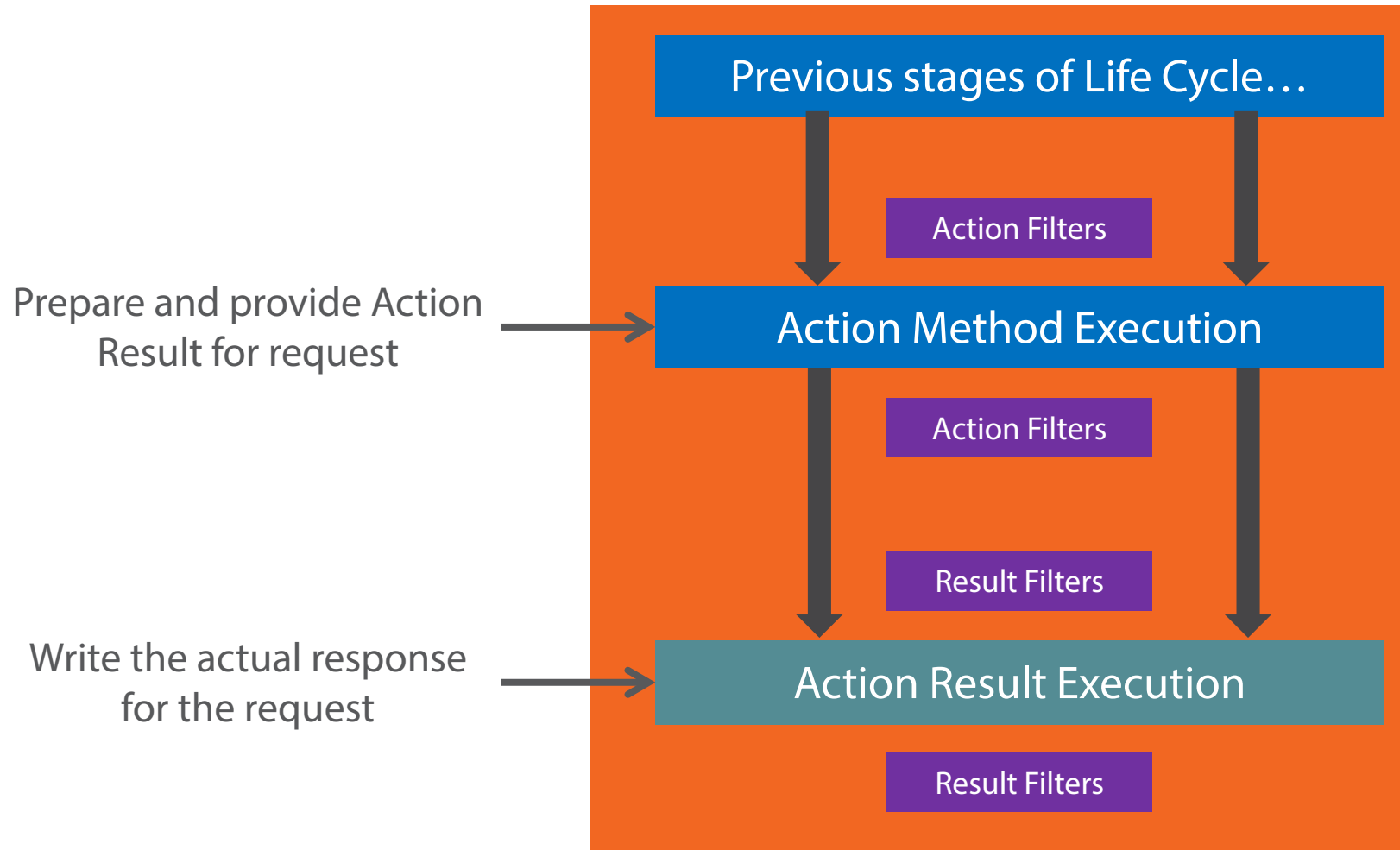
`return Content()`

`==`

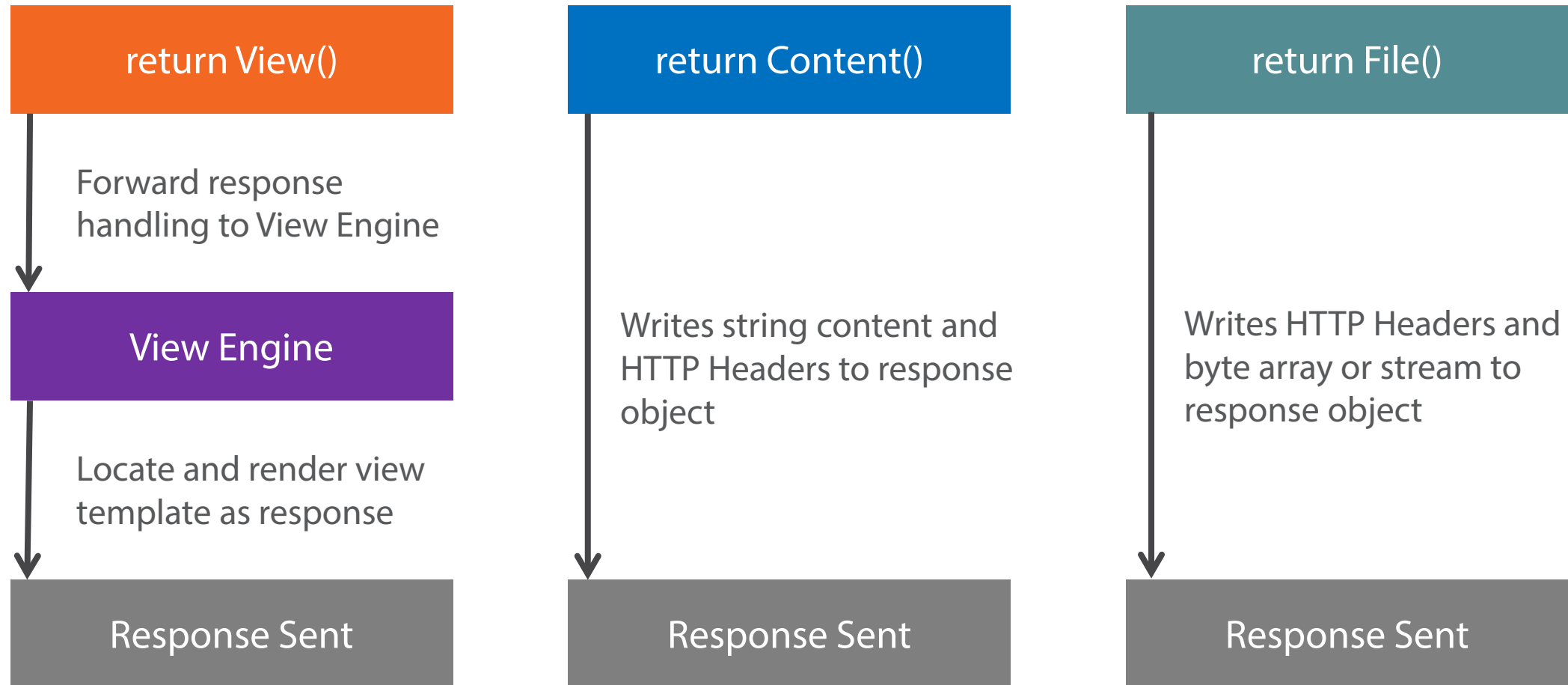
`return new ContentResult()`

Handling a Response with Action Results

(Partial) MVC Request Life Cycle

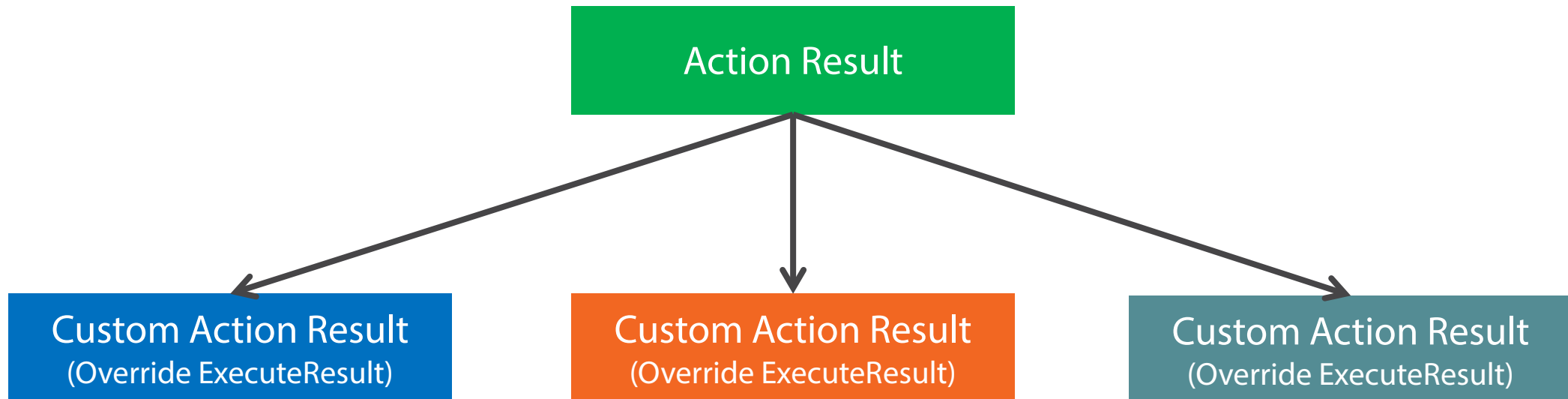


Understanding Different Result Types

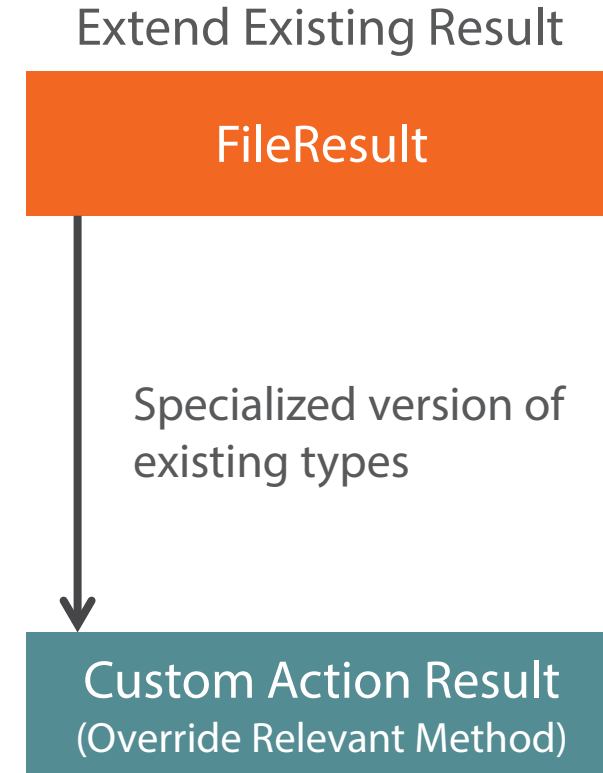
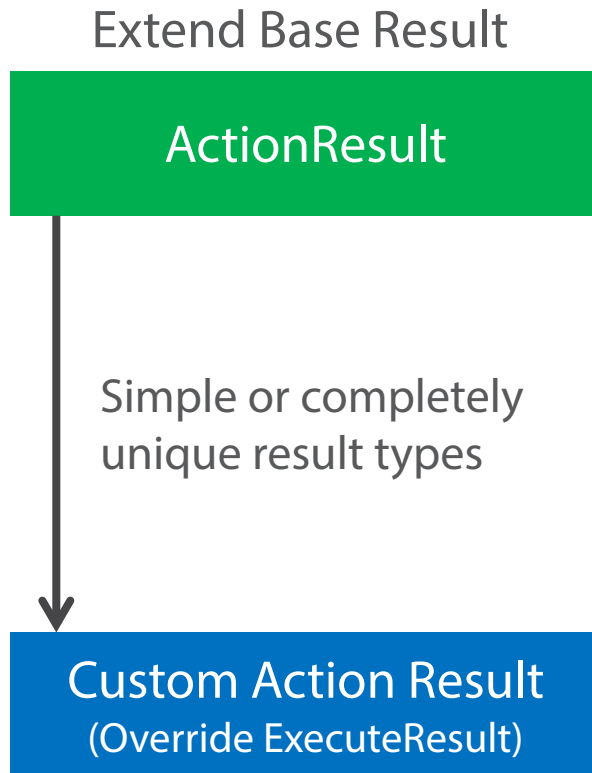


Building a Custom **Action Result**

```
public abstract class ActionResult
{
    public abstract void ExecuteResult(ControllerContext context);
}
```



Extending Existing Action Results



Summary



- Action Results are ultimately responsible for writing a response to a request
- Custom Results can be created by inheriting from ActionResult and overriding ExecuteResult
- Custom Results allow for cleaner Action Methods by abstracting code away behind the scenes
- Inherit from existing, more specific Action Results to reuse valuable logic