

Influencing Action Method Execution Using Custom Selectors



Alex Wolf

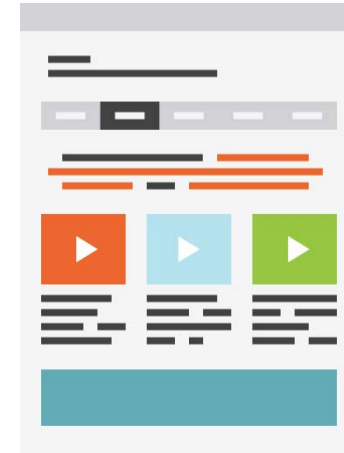
www.alexwolfthoughts.com

New Design, New Problems

Paladin Mobile App



Custom x-PaladinMobile
Http Header



New
Design



Old
Design



Action Method Selectors and Action Name Selectors

Working with **Selectors**

Action Name Selectors

`[ActionName("Home")]`

`public ActionResult Index()`

Action Method Selectors

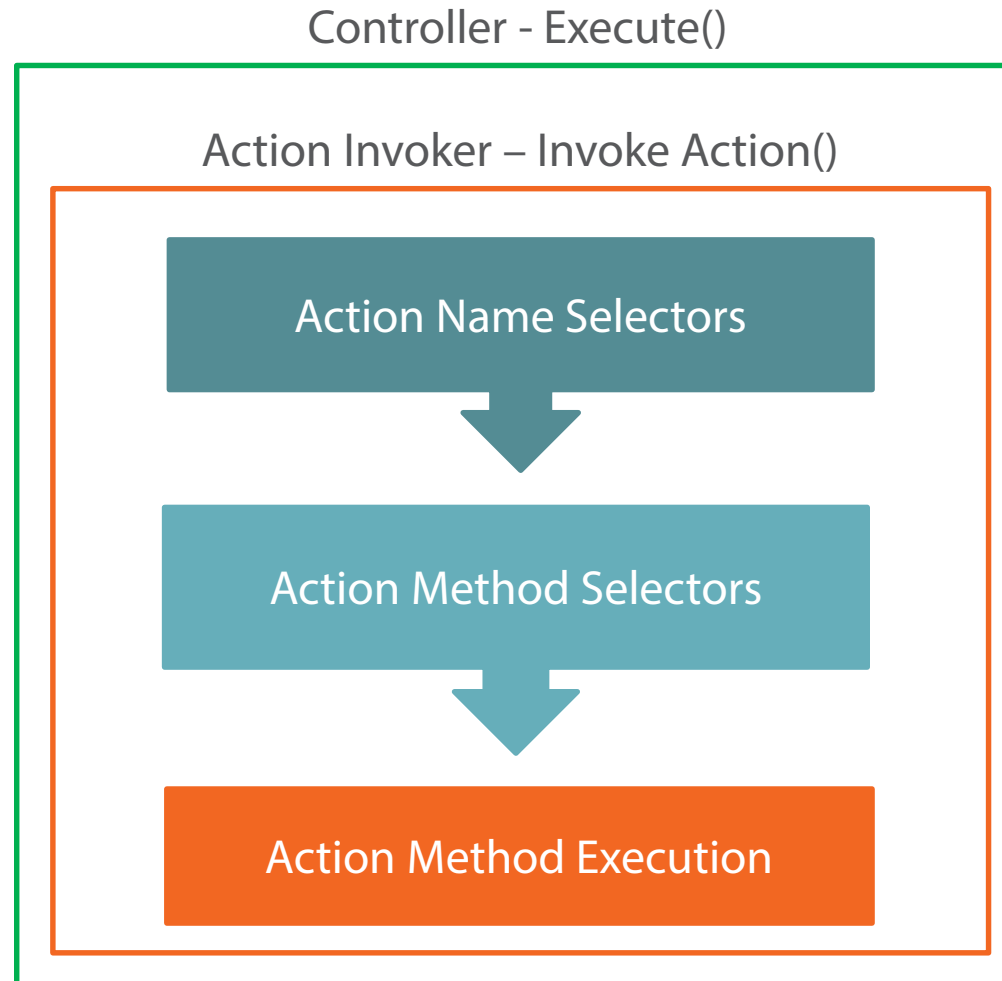
`[HttpGet]`

`public ActionResult Register()`

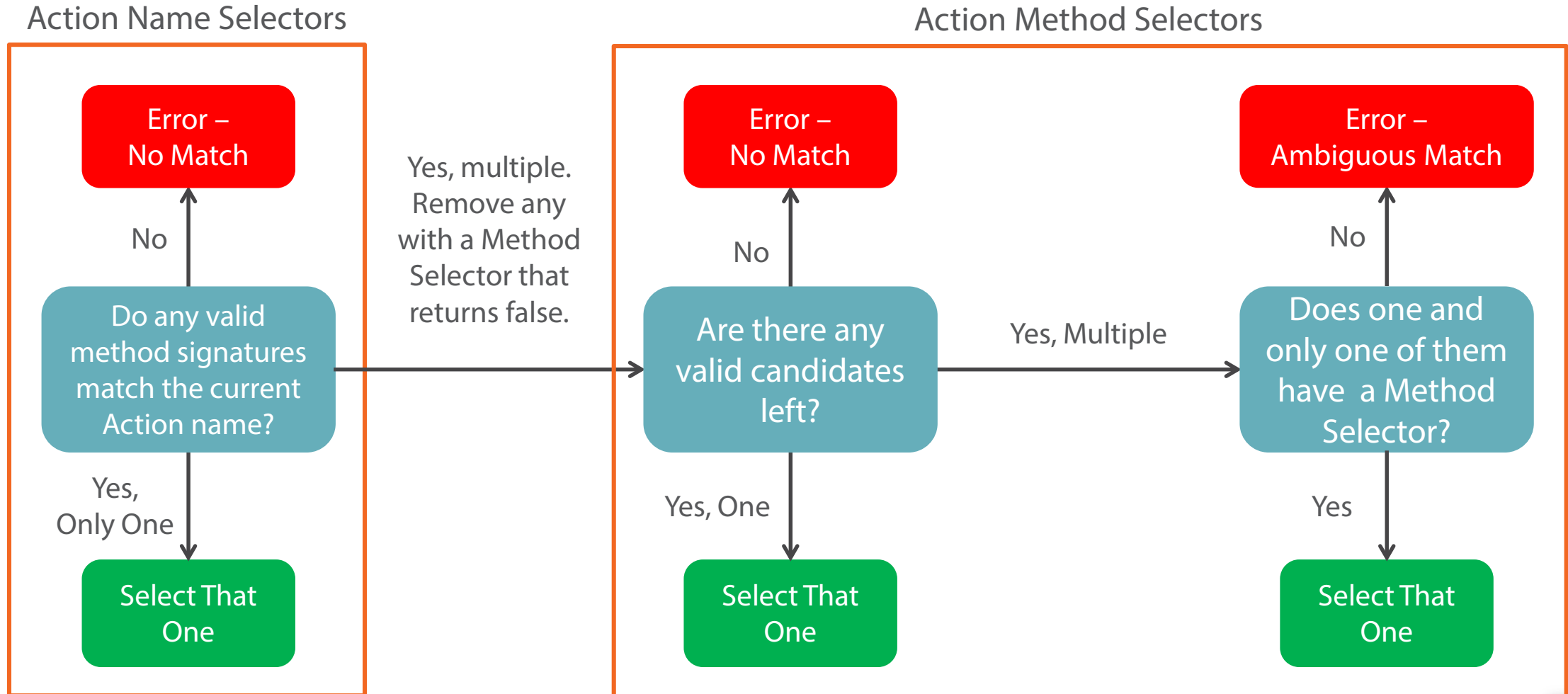
`[HttpPost]`

`public ActionResult Register()`

The Controller and Action Invoker



The Action Method Selection Process



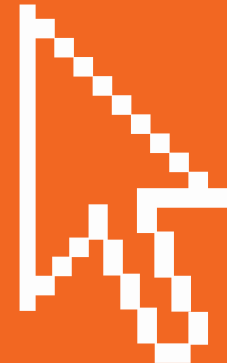
Extending Action Name and Action Method Selectors

```
public class NameSelector : ActionNameSelectorAttribute
{
    public override bool IsValidName(ControllerContext controllerContext,
        string actionName,
        MethodInfo methodInfo) { }
}
```

```
public class MethodSelector : ActionMethodSelectorAttribute
{
    public override bool IsValidForRequest(ControllerContext controllerContext,
        MethodInfo methodInfo) { }
}
```

Demo

Creating a Custom Action Method Selector



Summary



- Action Name and Method Selectors can influence the Action Invoker's behavior
- Custom implementations can be created through corresponding base attribute class
- Simple extension point with powerful influence
- Custom Controller base classes can abstract away reusable and miscellaneous data or methods