

# Struts Terminology

## Front Controller pattern

"The Front Controller consolidates all request handling by channeling requests through a single handler object. This object can carry out common behavior, which can be modified at runtime with decorators. The handler then dispatches to command objects for behavior particular to a request." [1]

"Front Controller channels all requests through a single *controller*. The controller itself is usually implemented in two parts: a handler and a hierarchy of *commands*." [3]

"The handler has two responsibilities:

- Retrieve parameters. The handler receives the HTTP Post or Get request from the Web server and retrieves relevant parameters from the request.
- Select commands. The handler uses the parameters from the request first to choose the correct command and then to transfers control to the command for processing.

The commands themselves are also part of the controller. The commands represent the specific actions as described in the Command pattern." [3]

<http://www.martinfowler.com/eaCatalog/frontController-sketch.gif>

Above: diagram by Martin Fowler

[http://msdn.microsoft.com/library/en-us/dnpatterns/html/Des\\_FrontController\\_Fig02.gif](http://msdn.microsoft.com/library/en-us/dnpatterns/html/Des_FrontController_Fig02.gif)

Above: diagram from MSDN pattern catalog

Struts is a Front Controller framework.

## Command pattern

"The Command pattern lets objects make requests of unspecified application objects by turning the request itself into an object. This object can be stored and passed around like other objects. The key to this pattern is an abstract Command class, which declares an interface for executing operations. In the simplest form this interface includes an abstract Execute operation." [2]

Struts employs Command pattern to process requests with command objects like Action and Command.

## Chain of Responsibility pattern

"Chain of Responsibility pattern allows to decouple senders and receivers by giving multiple objects a chance to handle a request. The request gets passed along a chain of objects until one of them handles it." [2]

"To forward the request along the chain, and to ensure receivers remain implicit, each object on the chain shares a common interface for handling requests and for accessing its successor on the chain." [2]

Struts employs Chain of Responsibility pattern to make request processing more flexible. Rather than subclassing a monolithic request processor object, it is possible just to replace, insert or remove Commands, if needed in order to extend or streamline the request processing flow.

## Command Object

A command represents an individual unit of work.

In Struts prior to 1.3 a command is implemented with an instance of Action class. Only one command can be defined per request. Chaining of commands is not supported. When commands are chained, the request processor code is executed once again, resulting in undesired behavior.

In Struts 1.3 a command can be implemented as an instance of Command class or as a series of Command classes combined into a chain, in addition to Action class. Chaining of Command objects is a recommended practice in Struts 1.3 intended to enhance the request processing flow.

After the command object finishes processing the request, it usually usually generates output by choosing which view to render. [3]

## Struts Terms

*Action class*, *Action*, *action* – Struts Action class, a request endpoint.

*JSP action*, *JSP action tag*, *action tag* – one of JSP tag elements: action, directive, declaration, expression, scriptlet, comment. Follows the XML format and always begins with <jsp:... />. JSP provides certain functionality through JSP actions such as finding or instantiating a JavaBean, or navigating to an URL. JSP actions are rarely called "actions" in context of Struts/JSP application to avoid ambiguity.

## References

- [1] [Front Controller, Martin Fowler](#)
- [2] [Design Patterns](#) - Design Patterns: Elements of Reusable Object-Oriented Software (Addison-Wesley)
- [3] [Front Controller Pattern, MSDN](#)