The `@WebFilter` annotation is used to declare a filter in a web application. The annotated class must extend the `javax.servlet.Filter` interface.

Syntax

```
@WebFilter(
    attribute1=value1,
    attribute2=value2,
    ...
)
public class TheFilter implements javax.servlet.Filter {
    // implements Filter's methods
}
```

Attributes

Name	Type	Required	Description
filterName	*String*	Optional	Name of the filter.
value or **urlPatterns**	*String[]*	**Required**	Specify one or more URL patterns to which the filter applies. Either of attribute can be used, but not both.
dispatcherTypes	*DispatcherType[]*	Optional	Specify types of dispatcher to which the filter applies. Default is `javax.servlet.DispatcherType.REQUEST`
servletNames	*String[]*	Optional	Specify names of servlets to which the filter applies.
displayName	*String*	Optional	Display name of the filter.
description	*String*	Optional	Description of the filter.
asyncSupported	*boolean*	Optional	Specify whether the filter supports asynchronous operation mode. Default is false.
initParams	*WebInitParam[]*	Optional	Specify one or more initialization parameters of the filter. Each parameter is specified by `@WebInitParam` annotation type.
smallIcon	*String*	Optional	Specify name of the small icon of the filter.
largeIcon	*String*	Optional	Specify name of the large icon of the filter.

Examples

- The following example registers a filter for the URL pattern */admin*:

```
@WebFilter("/admin")
public class MyFilter implements Filter {
    // implements Filter's methods here...
}
```

- Apply a filter for all URLs:

```
@WebFilter("/*")
public class MyFilter implements Filter {
    // implements Filter's methods here...
}
```

- Register a filter for a specific servlet:

```
@WebFilter(servletNames = "MyOwnServlet")
public class MyFilter implements Filter {
    // implements Filter's methods here...
}
```

- Register a filter for multiple servlets:

```
@WebFilter(servletNames = {"MyOwnServlet", "UploadServlet"})
public class MyFilter implements Filter {
    // implements Filter's methods here...
}
```

- Specify initialization parameters for the filter:

```
@WebFilter(
        urlPatterns = "/uploadFilter",
        initParams = @WebInitParam(name = "fileTypes", value = "doc;xls;zip;txt;jpg;png;gif")
)
public class UploadFilter implements Filter {
    // implements Filter's methods here...
}
```

- Specify additional information for the filter:

```
@WebFilter(
        urlPatterns = "/admin/*",
        filterName = "AdminFilter",
        description = "Filter all admin URLs"
)
public class MyFilter implements Filter {
    // implements Filter's methods here...
}
```

- Specify dispatcher types:

```
@WebFilter(
        urlPatterns = "/admin",
        dispatcherTypes = {DispatcherType.REQUEST, DispatcherType.FORWARD}
)
public class MyFilter implements Filter {
    // implements Filter's methods here...
}
```