```
ExtensionMethod ::= any token as defined by RFC 2616
        (that is, 1*[any CHAR except CTLs or separators])

HTTPMethod ::= "Get" | "POST" | "PUT" | "DELETE" | "HEAD" |
        "OPTIONS" | "TRACE" | ExtensionMethod

HTTPMethodList ::= HTTPMethod | HTTPMethodList comma HTTPMethod

HTTPMethodExceptionList ::= exclaimationPoint HTTPMethodList

HTTPMethodSpec ::= emptyString | HTTPMethodExceptionList |
        HTTPMethodList

transportType ::= "INTEGRAL" | "CONFIDENTIAL" | "NONE"

actions ::= null | HTTPMethodSpec |
        HTTPMethodSpec colon transportType
```

If duplicates occur in the HTTPMethodSpec they must be eliminated by the permission constructor.

An empty string HTTPMethodSpec is a shorthand for a List containing all the possible HTTP methods.

If the HTTPMethodSpec contains an HTTPMethodExceptionList (i.e., it begins with an exclaimation-Point), the permission pertains to all methods except those occuring in the exception list.

An actions string without a transportType is a shorthand for a actions string with the value “NONE” as its TransportType.

A granted permission representing a transportType of “NONE”, indicates that the associated resources may be accessed using any conection type.

Parameters:

`name` - the URLPatternSpec that identifies the application specific web resources to which the permission pertains. All URLPatterns in the URLPatternSpec are relative to the context path of the deployed web application module, and the same URLPattern must not occur more than once in a URLPatternSpec. A null URLPatternSpec is translated to the default URLPattern, “/”, by the permission constructor. All colons occuring within the URLPattern elements of the URLPatternSpec must be represented in escaped encoding as defined in RFC 2396.

`actions` - identifies the HTTP methods and transport type to which the permission pertains. If the value passed through this parameter is null or the empty string, then the permission is constructed with actions corresponding to all the possible HTTP methods and transportType “NONE”.

WebUserDataPermission(`java.lang.String` urlPatternSpec, `java.lang.String[]` HTTPMethods, `java.lang.String` transportType)

```
public WebUserDataPermission(java.lang.String urlPatternSpec, java.lang.String[]
        HTTPMethods, java.lang.String transportType)
```

Creates a new WebUserDataPermission with name corresponding to the URLPatternSpec, and actions composed from the array of HTTP methods and the transport type.

Parameters:

`urlPatternSpec` - the URLPatternSpec that identifies the application specific web resources to which the permission pertains. All URLPatterns in the URLPatternSpec are relative to the context path of the deployed web application module, and the same URLPattern must not occur more than once in a URLPatternSpec. A null URLPatternSpec is translated to the default URLPattern, “/”, by the permission constructor. All colons occuring within the URLPattern elements of the URLPatternSpec must be represented in escaped encoding as defined in RFC 2396.

`HTTPMethods` - an array of strings each element of which contains the value of an HTTP method. If the value passed through this parameter is null or is an array with no elements, then the permission is constructed with actions corresponding to all the possible HTTP methods.

`transportType` - a String whose value is a transportType. If the value passed through this parameter is null, then the permission is constructed with actions corresponding to transportType "NONE".

WebUserDataPermission(`javax.servlet.http.HttpServletRequest` request)

```
public WebUserDataPermission(javax.servlet.http.HttpServletRequest request)
```

Creates a new WebUserDataPermission from the HttpServletRequest object.

Parameters:

`request` - the HttpServletRequest object corresponding to the Servlet operation to which the permission pertains. The permission name is the substring of the requestURI (HttpServletRequest.getRequestURI()) that begins after the contextPath (HttpServletRequest.getContextPath()). When the substring operation yields the string "/", the permission is constructed with the empty string as its name. The constructor must transform all colon characters occuring in the name to escaped encoding as defined in RFC 2396. The HTTP method component of the permission's actions is as obtained from HttpServletRequest.getMethod(). The TransportType component of the permission's actions is determined by calling HttpServletRequest.isSecure().

Methods

equals(`java.lang.Object` o)

```
public boolean equals(java.lang.Object o)
```

Checks two WebUserDataPermission objects for equality. WebUserDataPermission objects are equivalent if their URLPatternSpec and (canonicalized) actions values are equivalent. The URLPatternSpec of a reference permission is equivalent to that of an argument permission if their first patterns are equivalent, and the patterns of the URLPatternList of the reference permission collectively match exactly the same set of patterns as are matched by the patterns of the URLPatternList of the argument permission.

Two Permission objects, P1 and P2, are equivalent if and only if P1.implies(P2) && P2.implies(P1).

Overrides: `equals` in class `Permission`

Parameters:

`o` - the WebUserDataPermission object being tested for equality with this WebUserDataPermission.

Returns: true if the argument WebUserDataPermission object is equivalent to this WebUserDataPermission.

getActions()

```
public java.lang.String getActions()
```

Returns a canonical String representation of the actions of this WebUserDataPermission. The canonical form of the actions of a WebUserDataPermission is described by the following syntax description.