

Goals / Use Cases

- Provide common interface for parameter handling on URLs and responses
 - Makes it easier to write state handling code
- Provide clear separation between the render parameters, which represent portlet state, and the action or resource parameters, which provide input for execution of a single request
- Improve usability of the portlet parameter handling interfaces
 - Increase regularity of the parameter handling method definitions and behavior across the four requests – render, action, resource, and event
 - Reduce the number of method calls necessary to handle state in a portlet
 - Resulting method definitions should behave intuitively. The developer should be able to understand the interfaces from the apidocs alone. A separate document describing the exceptions and special handling should not be necessary.
 - Eliminate the need to define the same method in multiple interfaces to avoid duplicating descriptions

7-1

JIRA Issues Addressed

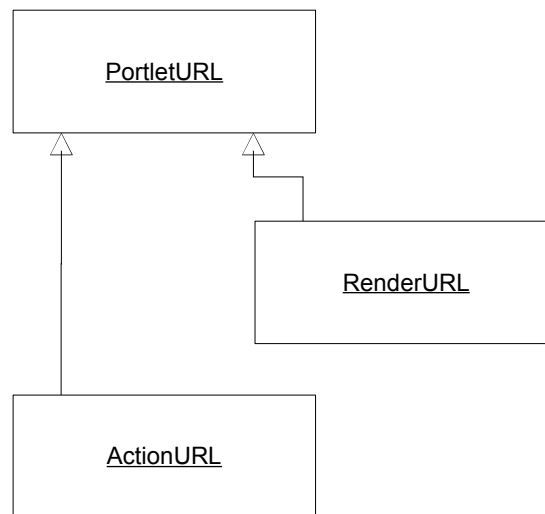
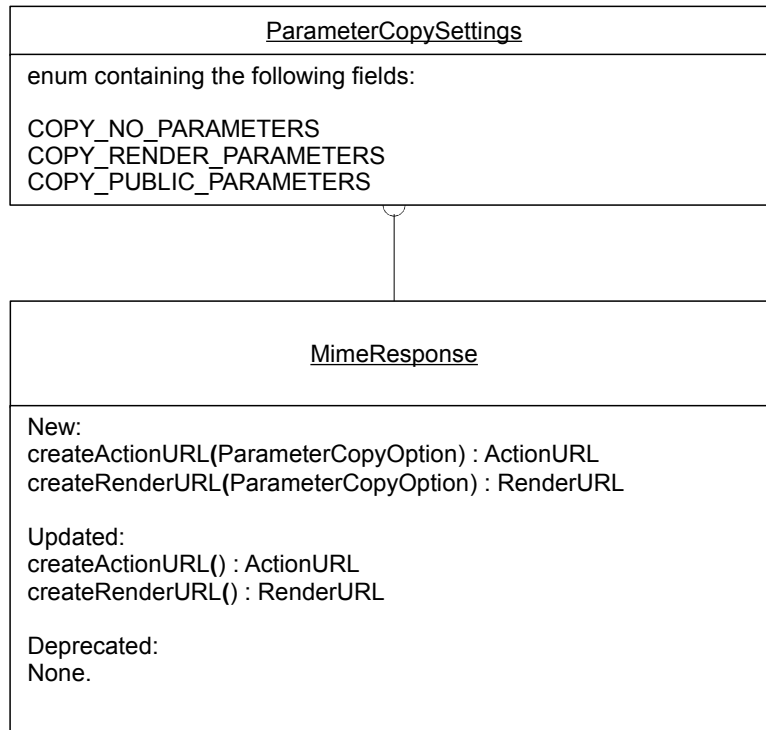
7-2

- Contains several proposals that can be considered independently
- Introduce new methods for URL creation that allow parameters to be automatically added - PORTLETSPEC3-30
 - Current URL creation method parameter handling unchanged
- Introduce idea of “persistent portlet render parameters” for the event and action phases as default for version 3.0 portlets, but controlled by a runtime option to allow for backward compatibility - PORTLETSPEC3-31
 - Portlet render parameters are persistent during the event and action phases and need only be explicitly set if they are to be changed.
 - Portlet render parameters are transported to the action request through the action URL.
 - Does not affect render phase or resource phase parameter handling.
- Introduce common parameter handling methods on requests and URLs - PORTLETSPEC3-11
 - Provide a common, regular interface for state handling
 - Cleanly separate Action & Resource Parameters from Render parameters

7-3:7-6



URL Creation Methods



ParameterCopyOption is an enum that defines values specifying how parameters are to be copied when a new URL is created. It contains the following fields:

COPY_NO_PARAMETERS – no parameters would be copied; same behavior as the current createActionURL() and createRenderURL() methods.

COPY_RENDER_PARAMETERS – the currently active portlet render parameters, including public render parameters, are copied to the URL.

COPY_PUBLIC_PARAMETERS – only the currently active public render parameters are copied to the URL.

Two new create methods are to be added to MimeResponse. The new methods each take a ParameterCopyOption enum as a parameter.

The current createActionURL and createRenderURL methods that do not take a parameter will not add parameters to the URL.

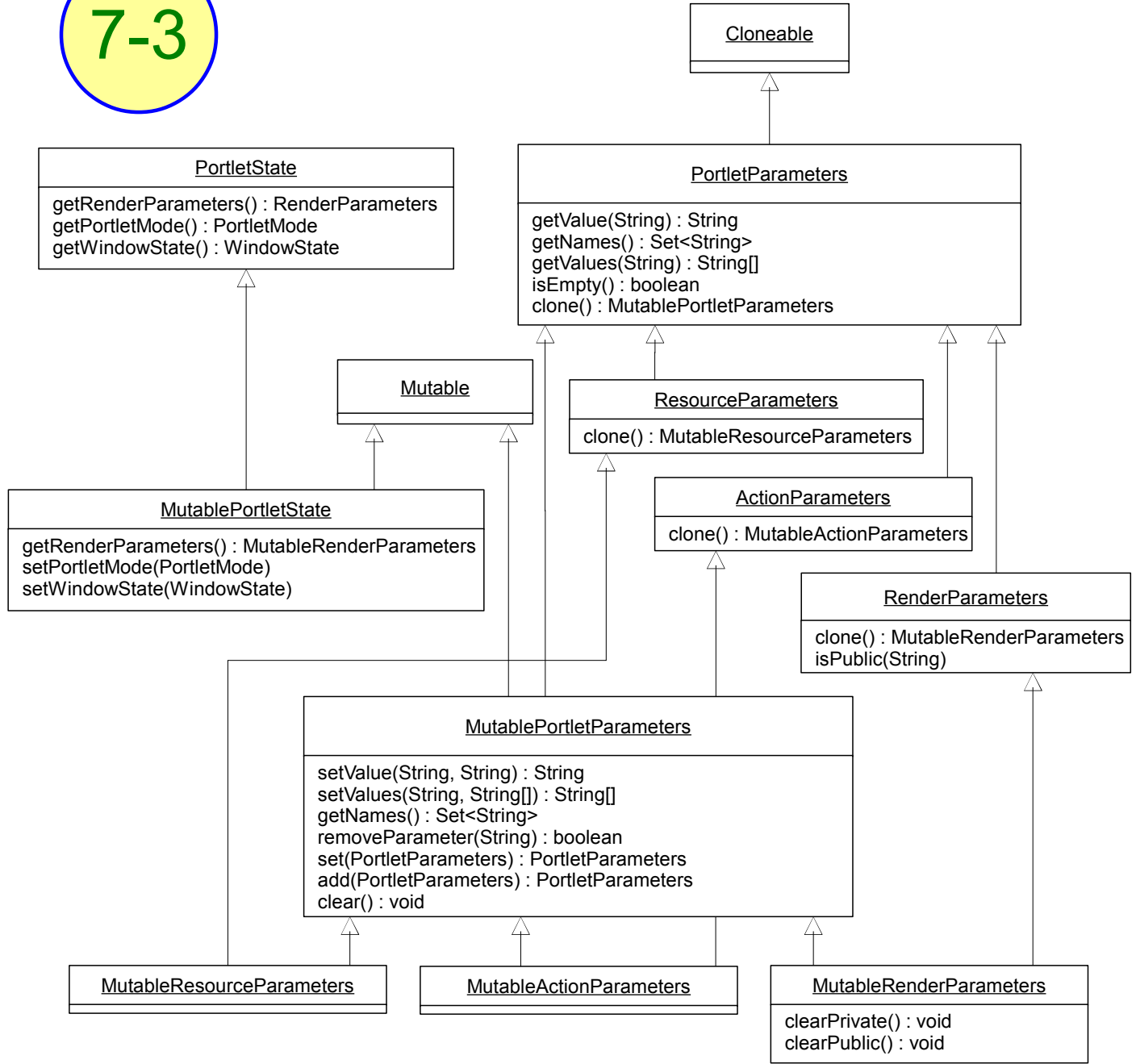
The createResourceURL method already copies the current portlet render parameters to the URL. These portlet render parameters on the URL cannot be changed. This behavior would remain the same, so no new createResourceURL method would be needed.

In addition, in order to best accommodate the new state handling interface presented on the following pages, two new interfaces will be introduced so that each URL type is represented by its own specific interface.

The interfaces would be introduced in a manner that would allow existing code that expects a PortletURL to be returned by createRenderURL() or createActionURL() to continue to work.

7-3

New Interfaces



PortletState #7: Extension of variant #6. Define a regular interface for parameter handling for portlet requests, responses, and URLs.

PortletState provides a common interface for URLs and the StateAwareResponse for obtaining the render parameters, window state and portlet mode.

PortletParameters represents the read-only portlet parameters such as those available from every portlet request. It is cloneable to allow for use cases that call for setting a similar set of parameters on many new URL objects.

Mutable is a marker interface designating methods that change portlet state.

ActionParameters and **ResourceParameters** are marker interfaces to improve type safety and to emphasize the resource parameter and action parameter concepts.

MutablePortletState extends PortletState with methods for setting the state.

RenderParameters provides an additional method for testing if a parameter is public.

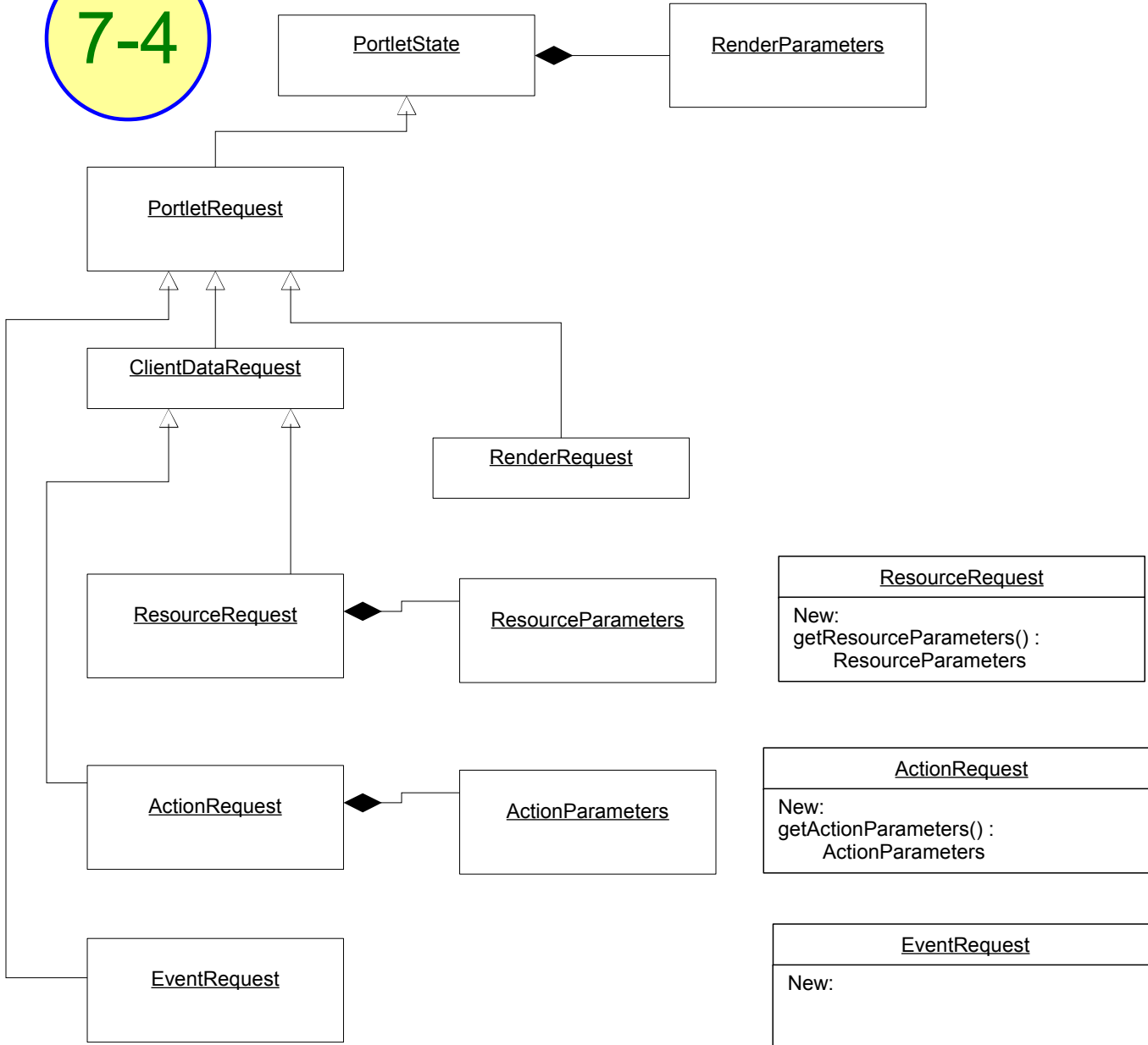
MutablePortletParameters extends PortletParameters by adding methods for setting and removing parameters. The clear() method clears all parameters. The set() method replaces all parameters with the values from the input PortletParameters object. After the set() operation is carried out, the target parameters object will contain precisely same parameter name-value pairs as the input object – no more and no fewer. The values are copied, so a change to the source PortletParameters object after the set() operation has been carried out will not cause a change to the parameters that were set in the target object. If the target MutablePortletParameters object was retrieved from a response or a URL, changes are immediately active. The new state will be available during subsequent requests as defined by the response or URL object.

MutableRenderParameters provides methods for explicitly clearing either public or private render parameters.

MutableActionParameters & **MutableResourceParameters** are marker interfaces to improve type safety and to emphasize the resource parameter and action parameter concepts.

7-4

Requests



<u>ResourceRequest</u>
New: getResourceParameters() : ResourceParameters

<u>ActionRequest</u>
New: getActionParameters() : ActionParameters

<u>EventRequest</u>
New:

PortletRequest:
extends the PortletState interface to provide access to the render parameters, portlet mode and window state.

getRenderParameters returns a read-only object representing the portlet render parameters, including the public render parameters.

ClientDataRequest:
adds no new parameter handling methods.

RenderRequest:
adds no new parameter handling methods.

getRenderParameters returns the portlet render parameters as they were last set during Action Phase or Event Phase processing, but including updated values of any public render parameters modified by other portlets.

ResourceRequest:
getResourceParameters returns the resource parameters, which consists of additional information that was appended to the URL after its creation.

getRenderParameters returns the portlet render parameters set on the resource URL triggering the request.

ActionRequest:
getActionParameters returns the action parameters, which consists of additional information appended to the URL after creation, along with any parameters, such as form parameters, provided by the client.

getRenderParameters returns the portlet render parameters set on the action URL triggering the action request.

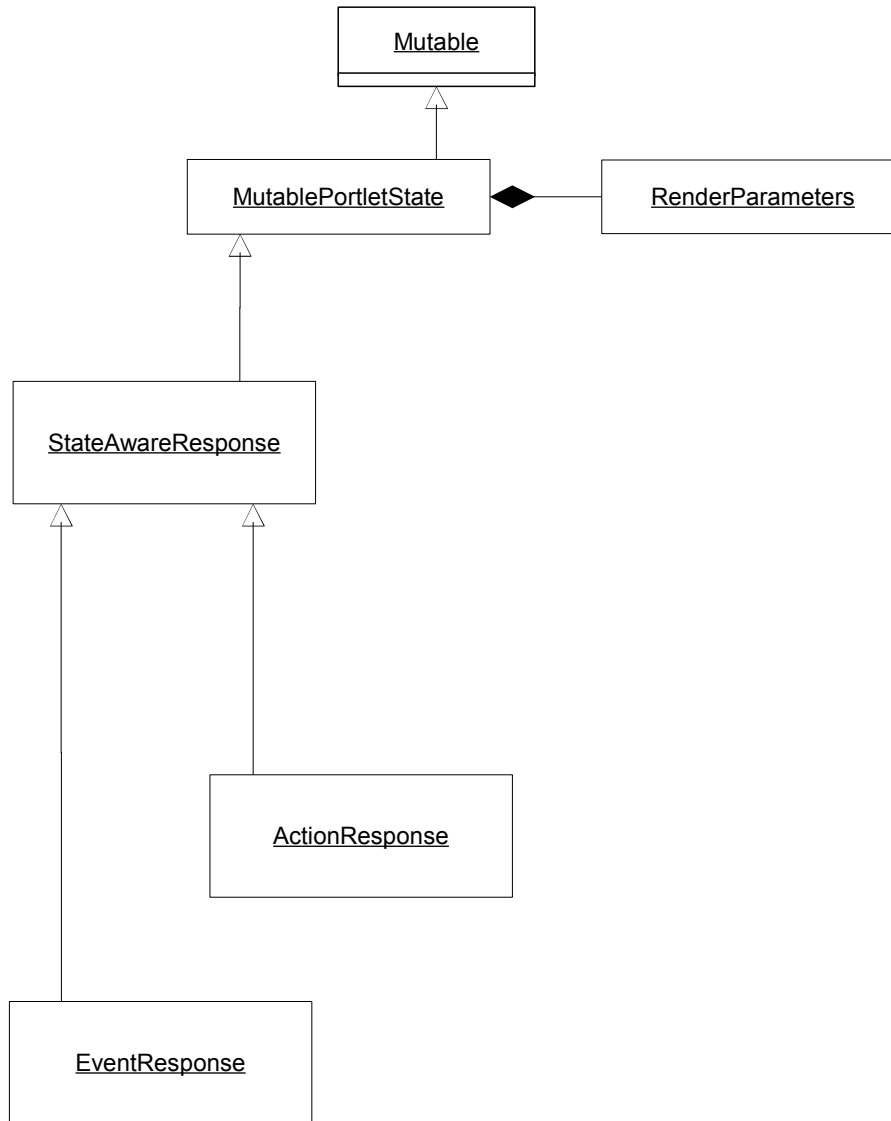
EventRequest:
adds no new parameter handling methods.

getRenderParameters returns the portlet render parameters as they were last set during Action Phase or Event Phase processing, but including updated values of any public render parameters modified by other portlets.

Note: The API enforces a strict separation between the action and render parameters and between the resource and render parameters. No methods for retrieving the combined action & render or resource & render parameters is provided.

7-5

Responses



Mutable:

marker interface designating methods that change portlet state.

MutablePortletState:

provides the portlet mode and window state getters and setters and method to obtain the render parameters to StateAwareResponse.

StateAwareResponse:

provides a new getRenderParameters() method through the MutablePortletState interface that returns an object methods to change the portlet parameters

The parameters returned depends on the state of the runtime option described in PORTLETSPEC3-31.

If the runtime option is set to provide compatibility to version 2.0 parameter handling, the MutablePortletParameters object returned by getRenderParameters is empty.

If the runtime option is set to allow version 3.0 parameter handling, the MutablePortletParameters object contains the same parameters returned by the the getRenderParameters() method on the EventRequest or ActionRequest.

ActionResponse:

provides no new parameter handling methods.

getRenderParameters returns the portlet render parameters set on the action URL triggering the action request (version 3.0 handling).

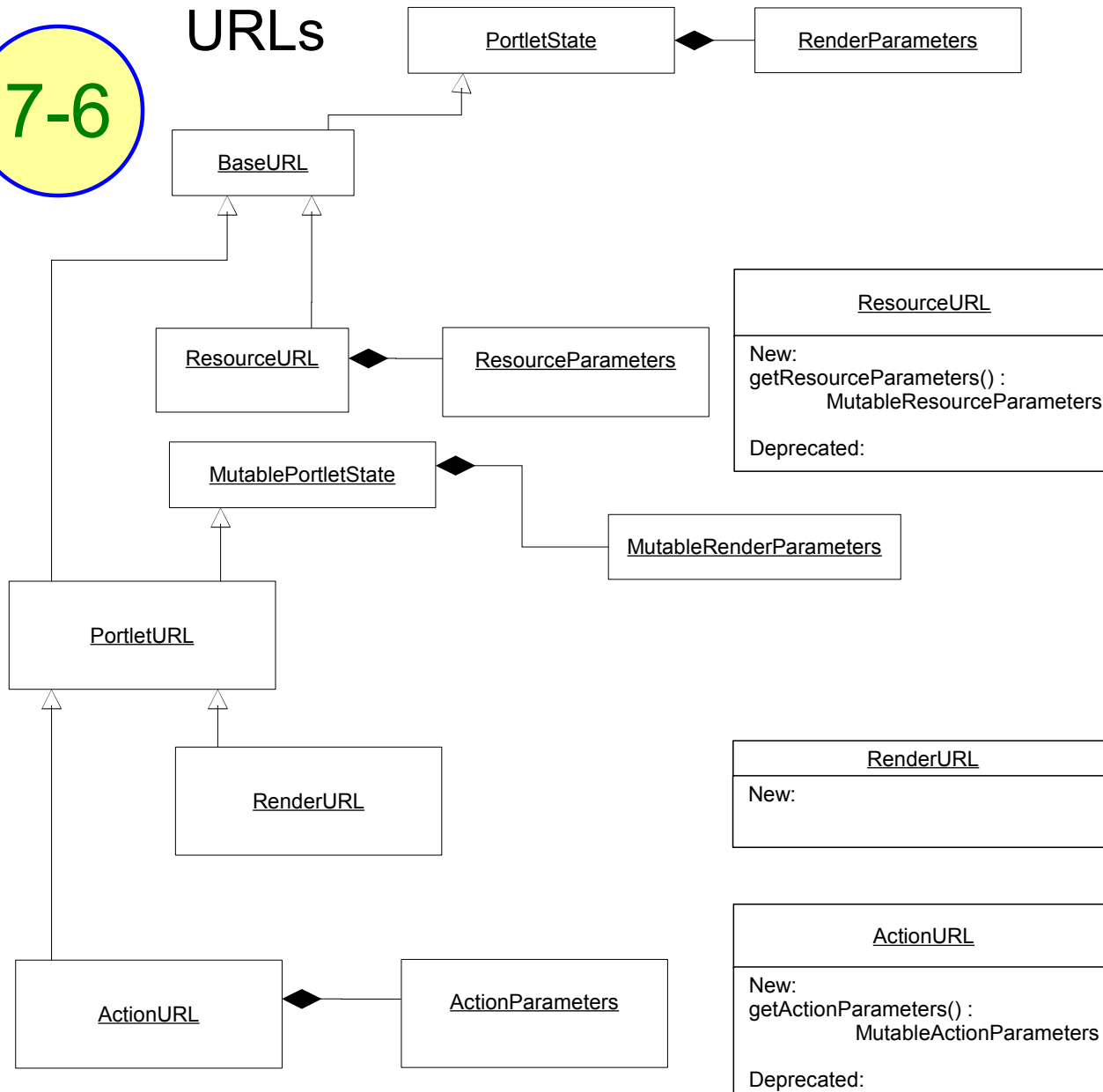
EventResponse:

provides no new parameter handling methods.

getRenderParameters returns the portlet render parameters as they were last set during Action Phase or Event Phase processing, but including updated values of any public render parameters modified by other portlets (version 3.0 handling).

7-6

URLs



BaseURL:

A new `getRenderParameters` method is added that returns a read-only object representing the portlet render parameters set on the URL.

ResourceURL:

An object representing the current portlet render parameters at URL creation time is automatically added to the `ResourceURL` when it is created. These parameters cannot be changed.

`getResourceParameters()` returns a mutable parameters object that allows resource parameters to be set. You cannot set or remove public render parameters through this object. Setting a parameter with the same name as a public render parameter will result in a resource parameter by that name.

PortletURL:

extends `MutablePortletState` to obtain the methods to get and set the render parameters, portlet mode and window state.

`getRenderParameters` is overridden to return a `MutablePortletParameters` object in order to allow the portlet render parameters to be changed on render and action URLs.

RenderURL:

abstracts the render URL.

No new parameter handling methods are provided.

ActionURL:

abstracts the action URL.

`getActionParameters()` returns a mutable parameters object that allows action parameters to be set. You cannot set or remove public render parameters through this object. Setting a parameter with the same name as a public render parameter will result in an action parameter by that name.