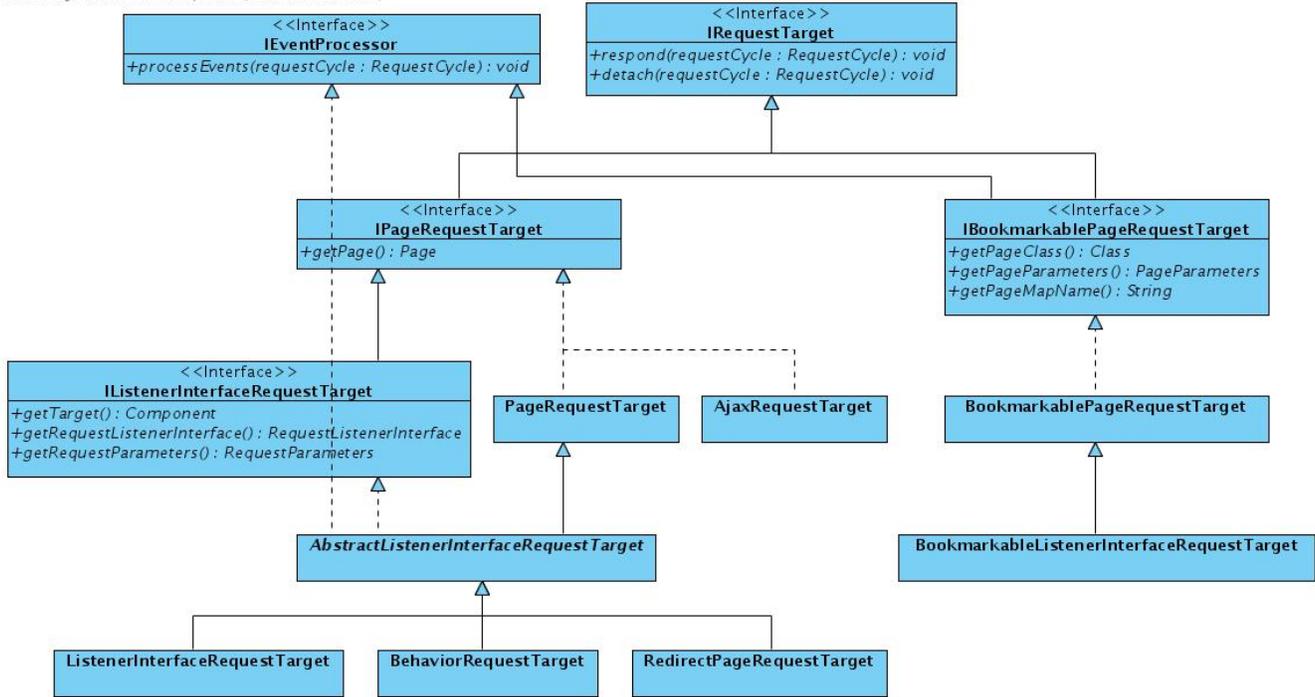


Request targets

Request target class hierarchy

Visual Paradigm for UML Community Edition (not for commercial use)



As it is said in javadoc request target is the base entity that is the subject of a request. Different types of request have different request targets. For instance a request for a bookmarkable page differs from a request for a link on a previously rendered page, which in turn differs from a request for a shared resource.

Request targets are created by IRequestCycleProcessor based on RequestParameters object which is translated from servlet request parameters. Properties of targets such as pageClass, pageParameter, page and others are put into targets by IRequestCycleProcessor. To see how these objects are obtained you should see implementation of IRequestCycleProcessor.

Targets have two main responsibilities:

- **process events** what means running user code which is bound to some events. Target can process events only if it implements IEventProcessor interface.
- **respond** what means generating some output in the form of HTML (or other markup language). Targets do not generate any response themselves but asks a Page to do it or a Component in case of AjaxRequestTarget.

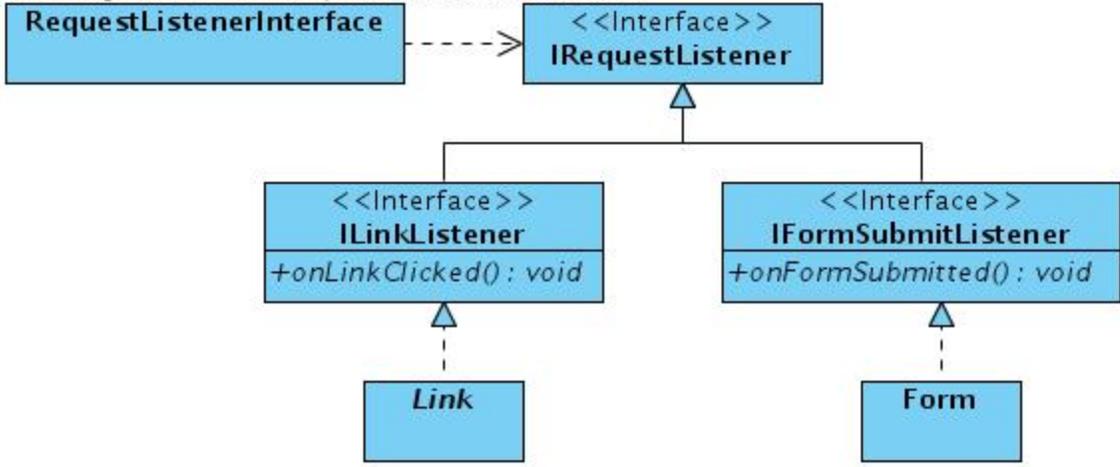
Probably the most commonly used targets are BookmarkablePageRequestTarget, PageRequestTarget and ListenerInterfaceRequestTarget.

BookmarkablePageRequestTarget stores class of the bookmarkable page and parameters with which that page can be created. BookmarkablePageRequestTarget is created every time request contains mounted URL or contains "wicket:bookmarkablePage" as a parameter (for example this kind of URL is generated by BookmarkablePageLink). Though it implements IEventProcessor this target does not actually process any events but creates a new bookmarkable page. That is for every request with mounted URL new bookmarkable page is created.

PageRequestTarget stores an instance of page and at rendering simply asks the page to render itself. This target is commonly created when you call code like setResponsePage(new MyPage()).

AbstractListenerInterfaceRequestTarget is the base class for targets which executes user code on some event. This target stores a component which has event handling code and an instance of **class** called RequestListenerInterface to which it delegates event processing.

There is one instance of **RequestListenerInterface** for all sub-interfaces of IRequestListener (which is a marker interface). Each of these instances stores a link (java.lang.reflect.Method) to the only method of the corresponding interface and can invoke this method on any component by RequestListenerInterface#invoke(Page, Component) method. So for example there is one instance of RequestListenerInterface for ILinkListener interface and it stores link to the onLinkClicked() method and this instance can attempt to call onLinkClicked() method on any component it is asked.



As other properties instance of RequestListenerInterface is put into a target by IRequestCycleProcessor implementation. When the target asks the instance of RequestListenerInterface to call an event handler, the target supplies this instance with page and component on which this call should be made. Then instance of RequestListenerInterface notifies the page and using reflection calls event handling method on the component.

ListenerInterfaceRequestTarget is very similar to AbstractListenerInterfaceRequestTarget. It is created when user clicks a link, submits a form or when selection changes for DropDownChoice which is not inside a form. Here is diagram of what ListenerInterfaceRequestTarget usually do to process events and respond.

