

Positron Configuration

Base Configuration

Positron's base configuration is provided by the "positron.json" file located in the "positron" folder at the root level of your application.

Override Configuration

Positron's override configuration, which provides configuration information that extends or overrides the base configuration, is provided by the "application.json" file, located at the root level of your application.

Configuration

Both Positron configuration files have the same structure. At any given level in the structure of the files, an element in the override configuration will replace that element in the base configuration. This allows the developer to replace a given implementation of an action or tag, etc, with a custom one.

Note that beyond the structure given by the base configuration, Positron does not apply any constraints to the configuration object. If the developer wishes to add further sections for the configuration of their application, then that is allowed and encouraged.

Configuration Keys

At the root level of the JSON object described by the composite (base & override) configuration files, there are certain fixed keys that Positron checks for. Some of these are direct key-value pairs, and others are key-composite pairs which describe a further level for subconfiguration.

The direct keys that Positron checks for are --

attributePrefix

This is the prefix that is applied to custom attribute names, to include attributes that are registered by Positron itself (such as *action* and *view*). HTML5 best practice suggests prepending "data-" to non-canonical attribute names, but this usage

varies.

This key is required, though it may be the empty string. The default value, provided by the base configuration, is "pos-".

cssClassPrefix

This is the prefix that is applied to Positron required CSS classes, such as *visible* and *invisible*. If this key is overridden, then CSS classes must be provided which are named consistently with its new value.

This key is required, though it may be the empty string. The default value, provided by the base configuration, is "pos-".

eventPrefix

This is the prefix that is applied by default to events dispatched by Positron actions, such as *ajax-form-success*.

This key is required, though it may be the empty string. The default value, provided by the base configuration, is "pos-".

tagPrefix

This is the prefix that is applied to custom tag names, to include tags that are registered by Positron itself (such as *if* and *list*, etc).

This key is required, though it may be the empty string. The default value, provided by the base configuration, is "pos-".

urlParameterPrefix

This is the prefix that is applied to reserved URL parameter names, which are currently only *log*.

This key is required, though it may be the empty string. The default value, provided by the base configuration, is "pos-".

pageCSSPath

This is the path, from the root of the application, that Positron uses to locate CSS files for Pages. The token "\$page;" is replaced with the "key" of the sought Page, when path resolution takes place.

This key is required. The default value, provided by the base configuration, is "pages/\$page;/\$page;.css".

pageHTMLPath

This is the path, from the root of the application, that Positron uses to locate HTML files for Pages. The token "\$page;" is replaced with the "key" of the sought Page, when path resolution takes place.

This key is required. The default value, provided by the base configuration, is "pages/\$page;/\$page;.html".

pageJSPath

This is the path, from the root of the application, that Positron uses to locate Javascript files for Pages. The token "\$page;" is replaced with the "key" of the sought Page, when path resolution takes place.

This key is required. The default value, provided by the base configuration, is "pages/\$page;/\$page;.js".

viewCSSPath

This is the path, from the root of the application, that Positron uses to locate CSS files for Views. The token "\$view;" is replaced with the "key" of the sought View, when path resolution takes place.

This key is required. The default value, provided by the base configuration, is "views/\$view;/\$view;.css".

viewHTMLPath

This is the path, from the root of the application, that Positron uses to locate HTML files for Views. The token "\$view;" is replaced with the "key" of the sought View, when path resolution takes place.

This key is required. The default value, provided by the base configuration, is "views/\$view;/\$view;.html".

viewJSPath

This is the path, from the root of the application, that Positron uses to locate

Javascript files for Views. The token "\$view;" is replaced with the "key" of the sought View, when path resolution takes place.

This key is required. The default value, provided by the base configuration, is "views/\$view;/ \$view;.js".

The compound keys that Positron checks for are --

actions

This key describes an object indirected from the base level configuration object that contains a list of key value pairs which configure the actions available to the application. The key in this object is the name of the action as referenced from markup; the value is the class name of the Javascript class which is to be instantiated to represent the action.

This key is required if any actions are available to the application. The default value is the Positron default action set. An example of an entry would be -

"addclass" : "positron.action.AddClassAction"

attributes

This key describes an object indirected from the base level configuration object that contains a list of key value pairs which configure the attributes available to the application. The key in this object is the name of the attribute as referenced from markup; the value is the class name of the Javascript class which is to be instantiated to represent the attribute.

This key is required if any attributes are available to the application. The default value is the Positron default attribute set. An example of an entry would be -

"pos-action" : "positron.attribute.ActionAttribute"

events

This key describes an object indirected from the base level configuration object that contains a list of key value pairs which configure the events available to the application. The key in this object is the name of the event; the value is the class name of the Javascript class which is to be instantiated to map the event into parameters.

This key is required if any events are available to the application. The default value is the Positron default event set. An example of an entry would be -

"click" : "positron.event.ClickEvent"

tags

This key describes an object indirected from the base level configuration object that contains a list of key value pairs which configure the tags available to the application. The key in this object is the name of the tag as referenced from markup; the value is the class name of the Javascript class which is to be instantiated to represent the tag.

This key is required if any tags are available to the application. The default value is the Positron default tag set. An example of an entry would be -

"pos-json" : "positron.tag.JSONTag"

triggers

This key describes an object indirected from the base level configuration object that contains a list of key value pairs which configure the triggers available to the application. The key in this object is the name of the trigger as referenced from markup; the value is the class name of the Javascript class which is to be instantiated to represent the trigger.

This key is required if any triggers are available to the application. The default value is the Positron default trigger set. An example of an entry would be -

"longtap" : "positron.trigger.LongTapTrigger"