

Grailsrocks Cheat Sheet

Platform Core

Full documentation: grailsrocks.github.com/grails-platform-core
For Platform Core 1.0.RC2 by Marc Palmer (marc@grailsrocks.com)

UI Extensions (controllers, tags)

`displayMessage(msg)` - `msg` -> view
`displayMessage(args)`
`displayFlashMessage(msg)`
`displayFlashMessage(args)`
`pluginSession` - Plugin's session vars
`pluginFlash` - Plugin's flash vars
`pluginRequestAttributes` - Plugin's request attributes

UI Extension Tags

`p:text` - `i18n` text with namespacing
`p:textScope` - Set `i18n` text scope
`p:label` - Render `<label>` with `i18n`
`p:button` - Render buttons with `i18n`
`p:displayMessage` - Render `msg` from
`p:smartLink` - `i18n` Link to a controller
`p:organization` - Render company name
`p:siteName` - Render website name
`p:siteURL` - Render website URL
`p:siteLink` - Render link to website
`p:year` - Party like it's `<p:year/>!`
`p:joinClasses` - Clean css class concat
`p:callTag` - Call another GSP tag by name
`p:attrIfSet` - Stop empty HTML attributes
`p:dummyText` - Lorem ipsum dolor sit amet

Events (controllers, services, domains)

`event(args, {})` - Send an event
`event(topic, {})`
`event(topic, data, {})`
`event(topic, data, params, {})`
`@Listener(topic)` - Define listener
`on(topic, {})`
`on(ns, topic, {})`
`waitFor(replies)` - Block for results
`removeListeners()`
`countListeners()`

Navigation Tags

`nav:primary` - Render primary menu
`nav:secondary` - Render secondary menu
`nav:menu` - Render any menu
`nav:items` - Iterate over menu items
`nav:set` - Set the current path or scope
`nav:activePath` - Get active path
`nav:activeNode` - Get active node
`nav:title` - Render a nav item title

The automatic config & `i18n` namespacing for plugins uses the prefix:
`plugin.<pluginNameHere>.`

Config (all artefacts)

`pluginConfig` - The plugin's config vars
... in plugin descriptor only:
`doWithConfig { }` - modify config
`doWithConfigOptions { }` - declare config options

Security Tags

`s:identity` - Return current user ID
`s:info` - Return auth plugin's user object
`s:ifLoggedIn`
`s:ifNotLoggedIn`
`s:ifPermitted`
`s:ifNotPermitted`
`s:createLogoutLink` - Render log out URL
`s:createLoginLink` - Render log in URL
`s:createSignupLink` - Render sign up URL
`s:logoutButton` - Render log out button
`s:loginButton` - Render log in button
`s:signupButton` - Render sign up button

Security (controllers, services, domains)

`securityIdentity` - Current user ID
`securityInfo` - Auth plugin's user object
`userExists(userId)`
`withUser(userId) { }` - Perform as user
`userHasAnyRole(list)`
`userHasAllRoles(list)`
`userIsAllowed(obj, perm)`