Content Security Policy (CSP)

For the NGClient we have on the admin page a number of properties that configures the CSP headers we set when a NGClient is launched.

For more information on CSP visit: Content Security Policy (CSP) - HTTP | MDN (mozilla.org).

You can disable it fully with the boolean property: servoy.ngclient.setContentSecurityPolicyHeader but this should be avoided.

the other properties:

servoy.ngclient.contentSecurityPolicy.frame-src:
servoy.ngclient.contentSecurityPolicy.frame-ancestors:
servoy.ngclient.contentSecurityPolicy.font-src:
servoy.ngclient.contentSecurityPolicy.img-src:
servoy.ngclient.contentSecurityPolicy.style-src:

Are there if you want to run the ngclient withing an iframe (frame-src, frame-ancestors) or want to relax certain behaviors even more for loading if styles/fonts/images

Servoy has defaults (see the info buttons on the admin page) so that the basic NGClient deployment works where most of the stuff are all coming from the same host "self"

Servoy only enables this for when we detect a browser that can handle CSP level 3, because we need for scripts/css the "strict-dynamic" property: strict-dynamic Explained (content-security-policy.com)

All scripts/links have a nonce which is also specified in the CSP header to allow all the scripts that Servoy generates from the web packages in the index.html (main solution index)

In NGClient1 we still allowed "eval" to happen, this is not allowed anymore for NGClient2, because of this you can't push functions as string to the client and eval it to get a function, if a component needs a property like that the property type needs to be "clientfunction" instead of "string" or "tagstring"

This way NGClient2 will generate a script from that on the server and that will be loaded as a normal js file.