

2020.12 Whats new

The developer is based on Eclipse 4.17 (2020.09) which now requires Java 11.

This is not a problem with our default installation because we ship and run the developer with Java 15.0.1

But developer using their own java, can't use Java 8 anymore to run Servoy 2020.12

Also upgraded a lot of other developer plugins libraries to be up to date to move our self to Java 11 later on.

Currently the runtime still supports Java 8 (so the WAR that you generate and deploy)

This will change in 2021 very likely the 2021.03 (LTS) will be the last that supports Java 8, so maybe 2021.06 or 2021.09 and later will fully require java 11

Various changes in the developer to handle better file changes that are changed outside of the Developer (like using an external git client)

NGDesktopExporter now exposes some new features for generating (update) packages (updates can only be generated with a new ngdesktop release after NGDesktop 2020.12)

You can generate packages for older versions if you want this.

NGDesktop 2020.12 will also be released build on the latest electron 11 release (Chrome 87). With support for updating it self (to a future NGDesktop 2021.x)

When installing the new 2020.12 ngdesktop users should first uninstall the old one.

Also the exporter needs a servoy cloud login to generate the artifacts, (so you need to login into your developer)

We generate a new warning for a missing return jsdoc tag (<https://support.servoy.com/browse/SVY-15473>)

This is because of our new parsing that you really should doc your function correctly so the validation and completion do work.

War export: The tomcat context options are removed and a file chooser for a context.xml file that needs to be included is added

You can now give a context file in the exporter where you can push in anything you want, this way you can configure any context property of tomcat you want
As an example `<CookieProcessor sameSiteCookies="none" />` adding that in the context file will support embedding ngclient inside another domain

databaseManager.getEditingRecords have a few different new overloads for filtering the edited records right away.

getEditingRecords(datasource:String)

getEditingRecords(datasource:String, filter:Object)

RestWS has a breaking change that can be turn back of with a setting

See <https://support.servoy.com/browse/SVY-15339> and <https://wiki.servoy.com/display/DOCS/RESTful+Web+Services> (search for 2020.12)

RestWS now support streaming files from the server to the client by returning a JSFile object instead of the byte[] directly

But it also supports now that files are uploaded to the server and it will use streaming for that so POST and PUT request will get JSUpload objects instead of directly the bytes.

This is default now and that's a breaking change, you can go back to the old behavior to set the property "rest_ws_use_jsupload_for_binary_data" to false

But do try to use the new behavior because for big files this will result in a lot less memory usage.

Do avoid then calling getBytes() on the JSUpload but save/stream it directly by using the write() function

The RestWS client plugin also got an option to not send the user properties as cookies

some smaller features:

> maxlength property on textarea (<https://support.servoy.com/browse/SVY-15086>)

> callback function parameter for a formpopup window is being closed (<https://support.servoy.com/browse/SVY-15423>)

> be able to block the close for a formpopup window (<https://support.servoy.com/browse/SVY-15330>)

> "Search for References" to Context Menu of In Mem Datasources (<https://support.servoy.com/browse/SVY-15549>)

Possible breaking changes

> see RestWS update above

> Column auto enter-> lookup method now gets the context (JSColumn object) as an argument

So if you reuse the lookup method that is attached to a column also in our solution with an option argument, that argument is now for the lookup that JSColumn

> We changed the way attributes work on components that are itself angular directives like the keylistener service.

So if you are implementing your own angular directive that depends on attributes on components you need to update that directive that it uses the \$svyAttributes service of servoy.

As an example: [Key Listener service](#) that listens for components with specific attributes.

If you implement your own (custom) components and you want to support such directive like keylistener you have to use the svy-attributes directive on components html: [SvyAttributes directive](#)

> By default Servoy will not query anymore for related data if the relation is a pkfk relation and the parent record is not yet stored in the database. This can be controlled by a property see: [Tuning the Server#Foundsetloadingandsavingtuning](#)

known issue

> if you use a repository_server that's not done in an inmemory table and you created that repository_server with a 2020.09 release then it could be that when upgrading to 2020.12 you will get a "upgrade repository" button on the admin page.

But that then bombs out with a primary key violation if that is the case you can fix that by running this statement on your database "repository_server" first:

update servoy_user_properties set property_value = 53 where name = 'repository_version'

you can add to be sure: 'and property_value = 52' after it (that should be the current value)

because you are already on a repo version 53 but the the whole system doesn't know that yet.

> solution uploads (.servoy file) to the admin page could result in a file already exist() error, quick fix for this is to use the file-upload.jar of an older (2020.09) version of servoy