

New in 7.0

Highlights

The Servoy 7 release introduced the Servoy Mobile Client. Servoy Mobile brings the Servoy paradigms of easy application development to the mobile arena.

Using Servoy Mobile it is easy to build Mobile Business applications that work offline, seamlessly synchronize data back and forth between the Mobile client and the backend systems and can be deployed both as native application or as webapp

To get started with Servoy Mobile, see the [Servoy Mobile documentation](#)

Servoy Developer

JSDoc

- Added `@this` annotation in JSDoc to type the 'this' keyword
- Added `@override` to annotate methods as overriding a method on a super form
- Added functions marked as constructor using the `@constructor` JSDoc tag as types in CodeCompletion globally. Note that it is not needed to access the type through its fully qualified path (so `scopes.xxxxx.MyType`). Instead just type the name of the function and when Code Completing, the `scopes.xxx` part of the path will be automatically inserted
- Added `@enum` support to mark an object containing a set of values as an enumeration, which can then be used in relations again
- Added the `*` type to be able to make a distinction between the lowest common denominator in JavaScript, which is Object and non-JavaScript types (for example Servoy's built-in types)
- Improved support for using types in JSDoc cross scopes. When a scope contains a function `MyType` which is marked as constructor, it can now be references inside the scope within JSDoc as just `MyType`, without prefixing it with the fully qualified path and outside the scope with the fully path, without the build system raising warnings that `MyType != scopes.xxxx.MyType`

Build system improvements

- Improved the build mechanism to resolve forward references in .js files and thus not generate builder markers
- Generate builder markers for usage of deprecated variables and methods outside of the scripting layer (Think table lookup value, column converters etc.)
- Generate builder markers for invalid calls to functions marked as constructor

Misc.

- Updated the Unit Test view to be able to:
 - rerun a failed test, or a subtree from a failed solution from the results view
 - double click (or right click-> go to file) on a test/form/scope to open the container script
 - view history of test runs and rerun any of them
- Support Code Completion of a single statement spread out over multiple lines

Solution Development



Behavior Changes

Servoy 7 introduces several behavior changes compared with previous Servoy versions:

- [SVY-3673](#) In Client Design mode, forms do not loose their Client Design state when hidden from the UI and then shown again
- [SVY-3787](#) In find mode or when using table/foundsetFilters, the value will be converted using the column's Column converter (if the column has one) before it is used in SQL. Before, the value was added 'as is' to the SQL which meant that the value had to be converted manually in scripting
- Due to renewal of our code-signing certificate, which comes with mandatory [OCSP](#) which is not supported by Java 5, the minimum Java version for Servoy 7 and up is Java 6

- Added placeholder support on TextFields, TextAreas & PasswordFields support in all clients (See <http://webdesignerwall.com/demo/html5-placeholder-text/> for an example what a Placeholder is)
- Added mnemonic support on Tabs of a TabPanel and Panes of an AccordionPane
- Added RightClick and DoubleClick event handling support to Client Design mode
- Added move cursor to movable elements in Client Design mode
- Added max length support on text fields through the field's format property
- Added support on text fields to specify in the formatter which characters are allowed to be entered, when using the '*' in the Display Format
- Added support to use values from a variable containing an object with values in relations. Example:
With the code below, in a relation the value of one of the status codes can be used, by using `STATUS_CODE.ACTIVE`

```

/**
 * @enum
 */
var STATUS_CODES = {
  ACTIVE: 1,
  INACTIVE: 2
}

```

- Added `databaseManager.getDataModelClonesFrom` function to make the functionality available in all clients (similar function was already available in the Maintenance plugin, but the maintenance plugin is to be used only inside Batch Processors)
- Added `JSDataSet.getColumnType`
- Added `JSFoundSet.allDataProviders`
- Added `databaseManager.getDataSource` for convenience
- Added orientation constants to `JSTabPanel`
- Added `JSRelation.FULL_JOIN` constant to be used with `QBSelect`
- Added `JSRelation.RIGHT_OUTER_JOIN` to be used with `QBSelect`
- Added `JSRelationItem.LITERAL_PREFIX` to be used with `QBSelect`
- Added `JSRecord.hasChangedData`
- Added `JSRecord.isNew`
- Added `JSTable.isMetadataTable`
- Added `JSTableObject.isMetadataTable`
- Added `plugins.headlessclient.getOrCreateClient` to get an existing client for a specified UUID or create and return a new client with the specified ID if it not already exists
- Added `executeAsyncRequest` to all request types of the HTTP plugin
- Added `sendBulkMail` to send out emails with the bulk mail Precedence header set to Bulk, to prevent getting replies from auto-reponders
- Added `QBWhereCondition` to allow modifying the where clause of `QBSelect` objects
- Added `concat` method on `QBAggregate`, `QBColumn`, `QBFunction` and `QBFunctions`
- Added `clear` method on `QBGroupBy`, `QBResult` and `QBSorts`
- Added `QBSelect.clearHaving`
- Added support for specifying the size of formPopups when showing the popup through `plugins.window.showFormPopup`
- Added `JSForm.removeMethod`
- Added `JSForm.removeVariable`
- Added `Utils.stringToNumber` variant with decimal support
- Added `databaseManager.addTableFilterParam` variants that take a `datasource` string as parameter
- Added parameter to `databaseManager.createDataSourceByQuery` variants that work with a sql string to specify the column types
- Added `UICONSTANTS.TABLEVIEW_WC_SCROLLABLE_KEEP_LOADED_ROWS`
- Added `JSCalculation.hasFlag` to retrieve is a calculation is of type UUID (annotated in JSdoc with `@type {UUID}`)
- Added `controller.showRecords` variants that specify the records using a `QBSelect` statement
- Made scopes lazy loading, to improve performance

Installation

- Added option to the last page of the installer to launch Servoy Developer