Base Language - Support #5120

Date/Datetime - from iso

01/29/2021 06:00 AM - Marian Edu

Status:	Closed		**Start date:**	
Priority:	Normal		**Due date:**	
Assignee:	Constantin Asofiei		**% Done:**	100%
Category:			**Estimated time:**	0.00 hour
Target version:				
billable:	No		**case_num:**	
vendor_id:	GCD			

Description

Related issues:	
Related to Base Language - Bug #5201: fix issues related to the error message...	**New**

History

#1 - 01/29/2021 07:27 AM - Constantin Asofiei

- Description updated

- Start date deleted (01/29/2021)

From Marian:

> Working on Json getDate/Datetime the date.parseIsoDate method is being used to parse the ISO formatted string back to a date. However, an invalid date string - invalid month/day - in 4GL will throw error while given the LENIENT resolver used in FWD will happily resolve that to a 'valid' date. Aka 2020-15-15 will return 2021-03-15, our tests shows error number 80 should be thrown because the month is invalid.

> Since in 4GL there isn't any from-iso-date function it's not clear if this method is the one to be fixed or we need something else for parsing dates in json (xml as well probably), the method is being called from a bunch of other places like rest and persist packages so maybe the current 'LENIENT' resolver was used in purpose and the method should stay as-is.

> Looks like Constantin added those methods so will just make this a direct call :)

> Thanks

#2 - 01/29/2021 07:35 AM - Constantin Asofiei

- Status changed from New to WIP

Marian, you are right, FWd is too lenient. I'm looking at it.

#3 - 01/29/2021 07:56 AM - Constantin Asofiei

Marian, the date.parseIsoDate usage in JsonArray (and maybe others) I don't think is enoubht (at least what I have in 3821c), as it is missing the

expected type for parsing this string.

For example, in getDate, I think it should allow a 2020-01-01T99:99:99.000-08:00 (even its time is invalid), while getDateTime will need to consider it. I haven't checked explicitly JsonArray or other objects, but instead I used write-json and read-json for a temp-table - 4GL allows this invalid string if the field is date, and rejects it with an error if the field is datetime.

Ovidiu: for read-json, in case of invalid string, I get this error (as an example):

```
Invalid hours in datetime value (11260)
Unable to convert JSON to native data type for field 'f1' in temp-table 'tt1'. (15363)
```

Did you make any changes related to these kind of errors?

A solution would be to add the expected BDT class as an argument to date.parseIsoDate and act accordingly.

After these changes, the REST/SOAP tests need to be checked, too.

#4 - 01/29/2021 08:37 AM - Ovidiu Maxiniuc

Constantin Asofiei wrote:

> Ovidiu: for read-json, in case of invalid string, I get this error (as an example):
> [...]
> Did you make any changes related to these kind of errors?

No. I did not.

> A solution would be to add the expected BDT class as an argument to date.parseIsoDate and act accordingly.

I really do not like the date.parseIsoDate() because it does the same thing multiple time, until it gets it right. If we have the expected date* BDT type maybe its fromIso(String) should be called directly. The method is protected but it can be made public to be accessible from serial package like this:

```
date d = new date();
d.fromIso(jsonReadValue);
```

#5 - 01/29/2021 08:46 AM - Marian Edu

Constantin Asofiei wrote:

> Marian, the date.parseIsoDate usage in JsonArray (and maybe others) I don't think is enoubht (at least what I have in 3821c), as it is missing the expected type for parsing this string.

Constantin, yes for date only the first part is being parsed - oddly enough using a '.' as separator works just fine too, then month and date values are validated like when you try to create a date with (yy, mm, dd). Otherwise I'm trying to see what the pattern is there but "2020-01-A" it gives also error that month must be between 1 and 12???

Base64 decode is next for json, there is a method in SecurityOps - checkFormat that seems to do the validation on base64 encoded string like in 4GL... might have to make that one public so we can use it before base64 decode which in java seems to work just fine in strings like "invalid" :(

#6 - 01/29/2021 09:59 AM - Constantin Asofiei

Ovidiu, the reason for date.parseIsoDate having fallback approaches is because a 2020-01-01 string can be parsed and read by 4GL's read-json when the target data-type is datetime or datetime-tz, but can't be parsed as an exact ISO datetime. This was the same reason for the LENIENT approach in the datetime.fromIso and others.

Same for a datetime or datetime-tz literal, when the target data-type is datetime - it will be parsed (assuming is valid).

#7 - 03/04/2021 02:14 PM - Constantin Asofiei

- % Done changed from 0 to 90

The changes are in 3821c rev 12081.

Also, rev 1010 of the xfer testcases project contains a set of tests of how literals are parsed as date/datetime/datetime-tz. See the date.literal.tests/ folder, use literals.json as input and jstest.p as the program. There are still some differences between the error messages raised in 4GL and FWD, plus some formatting of the string read by JsonArray.getjsontext (i.e. in 4GL is read as 2021\/02\/-26 while in FWD is read as 2021/02/-26).

What I found was that 4GL basically uses a YMD format following all the quirks for a normal 4GL literal. OTOH, for SOAP and REST, things are different: these really require ISO-8601 lenient parsing. So, I've added different APIs, one to be used by legacy SOAP/REST and others by FWD runtime.

#8 - 03/07/2021 10:17 AM - Greg Shah

What is left to do on this task?

#9 - 03/07/2021 10:23 AM - Constantin Asofiei

There are some differences in the error messages/codes reported by FWD in case of invalid literals, but the functional part (to parse it) works OK in my tests.

I guess we could move that to some other task.

#10 - 03/07/2021 10:51 AM - Greg Shah

Vladimir is making changes to the date parsing in #4880. Are the issues overlapping?

Are the tests broken because of the differences?

#11 - 03/07/2021 10:57 AM - Constantin Asofiei

Greg Shah wrote:

> Vladimir is making changes to the date parsing in #4880. Are the issues overlapping?

I think #4880 is more related to the date format than the actual date literal parsing.

> Are the tests broken because of the differences?

ERROR condition is raised correctly, just the message differs. The differences can be seen by comparing the .4gl and .fwd output files (they are not automated).

Also, these are not new issues, but already present in FWD.

#12 - 03/07/2021 11:35 AM - Greg Shah

OK, defer the work. Please create a task for it.

#13 - 03/08/2021 08:41 AM - Vladimir Tsichevski

Constantin Asofiei wrote:

> Greg Shah wrote:
>
>> Vladimir is making changes to the date parsing in #4880. Are the issues overlapping?
>
> I think #4880 is more related to the date format than the actual date literal parsing.
>
>> Are the tests broken because of the differences?
>
> ERROR condition is raised correctly, just the message differs. The differences can be seen by comparing the .4gl and .fwd output files (they are not automated).

I suppose the test is date.literal.tests/jstest.p. How can I build/run this test? Currently at the conversion time I got:

```
    [java] Failure in file 'date.literal.tests/jstest.p':
    [java] com.goldencode.ast.AstException: Error processing ./date.literal.tests/jstest.p
...
    [java] Caused by: java.lang.RuntimeException: Cannot find class/interface progress.json.objectmodel.jsona
rray in PROPATH.
```

#14 - 03/08/2021 10:02 AM - Greg Shah

These tests have OO 4GL code usage. You must checkout the [skeletons](#). The latest skeletons are on xfer. You can look at cfg/p2j.cfg.xml to see that the skeleton path is ./skeleton so you need to check out like this from the root of the testcases project:

bzr co sftp://vvt@xfer.goldencode.com/opt/fwd/skeleton

#15 - 03/15/2021 04:43 PM - Constantin Asofiei

- Related to Bug #5201: fix issues related to the error messages reported for invalid date/datetime/datetime-tz literals added

#16 - 03/15/2021 04:46 PM - Constantin Asofiei

Greg Shah wrote:

> OK, defer the work. Please create a task for it.

Added [#5201](#).

Marian: can you take a look at this part:

> plus some formatting of the string read by JsonArray.getjsontext (i.e. in 4GL is read as 2021\/02\/-26 while in FWD is read as 2021/02/-26). I don't think this is related to date, but it may be related to the / char.

#17 - 03/16/2021 03:05 AM - Marian Edu

Constantin Asofiei wrote:

> Marian: can you take a look at this part:
>
>> plus some formatting of the string read by JsonArray.getjsontext (i.e. in 4GL is read as 2021\/02\/-26 while in FWD is read as 2021/02/-26).
>> I don't think this is related to date, but it may be related to the / char.

Yes, this is related to character escapes in serialization and it's already fixed in #4384i.

#18 - 07/12/2022 06:41 AM - Greg Shah

- Status changed from WIP to Closed

- % Done changed from 90 to 100

I think this is complete and the remaining work is deferred to #5201.