Arvados - Story #16534

[controller] facilitate database access by localdb API methods

06/17/2020 03:51 PM - Tom Clegg

Status:	Resolved	**Start date:**	06/26/2020
Priority:	Normal	**Due date:**	
Assigned To:	Tom Clegg	**% Done:**	100%
Category:	API	**Estimated time:**	0.00 hour
Target version:	2020-07-15		

Description

Subtasks:

Task # 16536: Review 16534-testable	**Resolved**

Related issues:

Related to Arvados - Feature #16462: Expand arvados-controller to expose fore...	**New**

Associated revisions

Revision dddfa30b - 07/01/2020 03:56 AM - Tom Clegg

Merge branch '16534-localdb-postgresql'

refs #16534

Arvados-DCO-1.1-Signed-off-by: Tom Clegg <tom@tomclegg.ca>

Revision 67d3cbd9 - 07/13/2020 06:12 PM - Tom Clegg

Merge branch '16534-testable'

refs #16534

Arvados-DCO-1.1-Signed-off-by: Tom Clegg <tom@tomclegg.ca>

History

#1 - 06/17/2020 03:52 PM - Tom Clegg

- Related to Feature #16462: Expand arvados-controller to expose forecast features added

#2 - 06/24/2020 02:17 PM - Tom Clegg

- Status changed from New to In Progress

#3 - 06/26/2020 08:47 PM - Tom Clegg

With this change, an API method can use the database like this:

```
        tx, err := currenttx(ctx)
        if err != nil {
                return
        }
        // ...
        err = tx.QueryRowContext(ctx, "
select uuid, api_token, expires_at, scopes from api_client_authorizations where api_token=$1"
, tokensecret).Scan(&resp.UUID, &resp.APIToken, &exp, &scopes)
```

The transaction will be committed if the handler returns success, otherwise it will be rolled back.

The transaction is started only if/when someone calls currenttx(). This avoids wasted overhead in the (currently most common) case where an entire controller request is handled without using the database at all (e.g., requests that are handled entirely in rails and federation requests that don't touch the local db).

Test cases should do everything in a transaction, and rollback when finished. To do this, use testdb() and testctx() like in LDAPSuite:

```
type LDAPSuite struct {
        cluster *arvados.Cluster
        ctx     context.Context
        rollback func()
}

func (s *LDAPSuite) SetUpSuite(c *check.C) {
        s.cluster = // ...
        s.db = testdb(c, s.cluster)
}

func (s *LDAPSuite) SetUpTest(c *check.C) {
        s.ctx, s.rollback = testctx(c, s.db)
}

func (s *LDAPSuite) TearDownTest(c *check.C) {
        s.rollback()
}
```

16534-localdb-postgresql @ 2c5417221843491727e4e5505012fc115e3bc7b0 -- https://ci.arvados.org/view/Developer/job/developer-run-tests/1935/

There's lots of room for more development (like using something ORM-like instead of deserializing fields individually). This branch just focuses on making a db transaction available to any API method that needs one.

#4 - 06/29/2020 09:48 PM - Lucas Di Pentima

Sorry for the delay, I've been re-reading the code several times and auto-answering lots of questions, the ones unanswered were:

- File lib/controller/localdb/db.go
    - Not really a question -- Line 33: Func name on documentation (WithTransaction instead of Transaction)
    - How transaction.setup works? I see that there’s a couple of txn.setup.Do(func() {}) (lines 38 & 83) and as I understand sync.once.Do() just blocks while the first call is still running and the just ignores the rest.
    - Regarding the previous point: would it be possible/useful to test this race protection?

The rest LGTM.

#5 - 06/30/2020 09:05 PM - Tom Clegg

> - Not really a question -- Line 33: Func name on documentation (WithTransaction instead of Transaction)

Changed both to ContextWithTransaction

> - How transaction.setup works? I see that there’s a couple of txn.setup.Do(func() {}) (lines 38 & 83) and as I understand sync.once.Do() just blocks while the first call is still running and the just ignores the rest.

Added comments here, and fixed a "currenttx() might return tx==nil, err==nil" bug.

```
                txn.setup.Do(func() {

// Using (*sync.Once)Do() prevents a future

// call to currenttx() from opening a

// transaction which would never get committed

// or rolled back. If currenttx() hasn't been

// called before now, future calls will return

// this error.
                        txn.err = errors.New("
refusing to start a transaction after wrapped function already returned")
                })
```

> - Regarding the previous point: would it be possible/useful to test this race protection?

Yes! Added a test case.

16534-localdb-postgresql @ ff22ba71afe839832943099cc1fe273197c45ec7 -- https://ci.arvados.org/view/Developer/job/developer-run-tests/1938/

#6 - 06/30/2020 09:30 PM - Lucas Di Pentima

Provided the tests pass, 16534-localdb-postgresql LGTM. Thanks!

#7 - 07/01/2020 03:32 PM - Peter Amstutz

- Target version changed from 2020-07-01 Sprint to 2020-07-15

#8 - 07/06/2020 08:45 PM - Tom Clegg

16534-testable @ a2b994f10fd73bdd882e691854239fc2d3b2e3a0 -- https://ci.arvados.org/view/Developer/job/developer-run-tests/1945/

- Moves the "use context to propagate current transaction" code into its own package, ctrlctx (is that too many consonants?). With this, we can put a group of handlers (like "user methods") in its own package that gets imported by localdb, without creating a localdb→userpkg→localdb import cycle when the handlers try to use the database transaction.
- Adds arvadostest methods for such controller packages to use, so (assuming they don't involve railsapi) they can do the "begin tx; do a sequence of controller methods; check results; rollback" thing instead of calling database/reset.
- Uses sqlx.Tx instead of plain sql.Tx, so controller packages can use conveniences like StructScan.

#9 - 07/07/2020 05:23 PM - Lucas Di Pentima

Reviewing branch 16534-testable

- Can you explain why was RoutableFunc moved to the Go SDK? Is it something developers may need to interact with a cluster?
- File lib/ctrlctx/db.go Line 72: Should the example call New() instead of NewContext()?
- File lib/ctrlctx/db_test.go Line 38: Could that test ask for arvadostest.DB() like it’s done on lib/controller/localdb/login_ldap_test.go Line 95?
- At file lib/controller/localdb/login.go Line 134: Could we already use StructScan() from sqlx?

#10 - 07/07/2020 06:47 PM - Tom Clegg

> - Can you explain why was RoutableFunc moved to the Go SDK? Is it something developers may need to interact with a cluster?

Router tests import arvadostest, which imports ctrlctx to manage transactions, so ctrlctx can't import router without creating a cycle.

So far we only use it in server code. It could go in a (new?) package like controller/api?

> - File lib/ctrlctx/db.go Line 72: Should the example call New() instead of NewContext()?

Yes, fixed

> - File lib/ctrlctx/db_test.go Line 38: Could that test ask for arvadostest.DB() like it’s done on lib/controller/localdb/login_ldap_test.go Line 95?

No, that would create an import cycle because arvadostest imports ctrlctx to manage transactions.

> - At file lib/controller/localdb/login.go Line 134: Could we already use StructScan() from sqlx?

Perhaps, but the sqlx conveniences are meant to be optional so I don't want to block on that.

16534-testable @ 21d95b374fc204ae021948e70dd8cce264fda05c

#11 - 07/07/2020 07:14 PM - Lucas Di Pentima

Tom Clegg wrote:

> Router tests import arvadostest, which imports ctrlctx to manage transactions, so ctrlctx can't import router without creating a cycle.
> So far we only use it in server code. It could go in a (new?) package like controller/api?

Right, I think it'll be less confusing to have the server code outside the Go SDK, but maybe there're other *pros* I'm not seeing.

> > - At file lib/controller/localdb/login.go Line 134: Could we already use StructScan() from sqlx?
>
> Perhaps, but the sqlx conveniences are meant to be optional so I don't want to block on that.

Ok, I supposed it would be trivial to do and useful as an example for future contributors.

So, other than this comments, it LGTM.

#12 - 07/09/2020 03:09 PM - Tom Clegg

Moved the RoutableFunc type to lib/controller/api package so it doesn't clutter the client-side code.

16534-testable @ 7be768ebfe665bcb30f4212b3f211c97b2fd65b9 -- https://ci.arvados.org/view/Developer/job/developer-run-tests/1951/

#13 - 07/13/2020 05:14 PM - Lucas Di Pentima

Tom Clegg wrote:

> Moved the RoutableFunc type to lib/controller/api package so it doesn't clutter the client-side code.

Thanks!

> 16534-testable @ 7be768ebfe665bcb30f4212b3f211c97b2fd65b9 -- https://ci.arvados.org/view/Developer/job/developer-run-tests/1951/

This LGTM, please merge.

#14 - 07/15/2020 02:21 PM - Tom Clegg

- Status changed from In Progress to Resolved

#15 - 10/07/2020 02:11 AM - Peter Amstutz

- Release set to 25