Registry PE Authentication and Authorization

The Registry Authnz process starts in *RegistryAuthComponent* which injects a *beforeFilter* callback into all requests.

- For API requests, *beforeFilter* will perform authentication of the API user. If successful, authorization is processed.
- For UI requests:
  - Requests for *PagesController* are granted.
  - If there is no current authenticated user in the session
    - Control is passed to `/auth/login/login.php`, which is where web server authentication is triggered.
    - Once the web server completes authentication, `login.php` returns control to the *TrafficController process-login* action.
    - *TrafficController* records the login event. *TrafficController* is intended to eventually be the location for authentication plugins (CO -1983). Control is then returned to the page originally requested.
  - Once there is an authenticated user, authorization is processed.

API Users are bound to specific COs. For UI logins, the authenticated identifier is mapped to Identifiers flagged for login attached to Person records. A given identifier may be valid in more than one CO.

Authorization is performed in *RegistryAuthComponent* based on the rules established within each table. Each table sets an array of permissions using `Per missionsTrait::setPermissions`, either directly as an array or as a closure that returns an array. The array has two required keys, `entity` and `tab le`, and one optional key, `related`.

entity and *table* are each a hash where the keys are actions that operate over entities (ie: requiring an `$id`) or tables, respectively. The values are either `fa lse`, indicating that action is never permitted, or an array of roles representing who is authorized to perform the action. Defined roles are

- `coAdmin`
- `coMember`
- `platformAdmin`

related, if provided, is an array (list) of models whose permissions are required, typically for table views to render related actions.

When a closure is provide, the expected function signature of the closure is

`function (\Cake\Http\ServerRequest $r, \App\Controller\Component\RegistryAuthComponent $auth, ?int $id): array`

StandardController also provides the permissions calculated by *RegistryAuthComponent* to views.

See Also

- Registry User Roles
- Registry Identifiers
- REST API v2