User type

final, form, projection

User type (projection) was added in CzechIdM version 10.2.0. Projection defines frontend form to read, create and edit user. We can create and edit user by different form. For example externe and internal employee can be created and edited differently (different attributes has to be filled). Used projection for user creation is set as user type.

Configurable form in product

Form projection with configurable features was added into product `core` module:

- Frontend:
  - Form projection content is placed on path `src/content/identity/projection/IdentityProjection.js`.
  - Route (= frontent target) `form/identity-projection/:entityId' is registed in `routes.js`.
- Backend:
  - Route component `IdentityFormProjectionRoute` with route usage is registered.
  - `IdmIdentityProjectionDto` was added - contains all projection data sent between backend and frontend, contains:
    - **identity** - identity.
    - **contract** - first (~ prime) contract. If currenly logged user has permission to read prime contract. First other contract is shown otherwise. Contract are sorted by priority the same way, as prime contract is evaluated.
    - **otherContracts** - all other contracts.
    - **otherPositions** - all other positions.
    - **identityRoles** - all assigned identity roles (loaded by projection configuration property `load-assigned-roles` is enabled).
  - `IdentityProjectionManager` was added and support **get and save** identity **projection**. Projection is processed by event and processors (see `IdentityProjectionSaveProcessor`). Processors can be registed to process `IdmIdentityProjectionDto` content (custom validations, generators etc. can be added non invasively). Manager contains overridable protected method to get and save all parts of projection (methods can be overidden invasively to add or change product behavior).
  - `IdmIdentityProjectionController` is exposed on url '<server api>/identity-projection' and support get and post `IdmIdentityProjectionDto` - calls `IdentityProjectionManager`methods.

If projection works with role requests only, then loading of all assigned identity roles could be disabled - by projection configuration property `load-assigned-roles`

Default user detail

Default user detail is still available and is used for users without projection is specified. Default user detail can be used as projection with route `/form/identity` too:

- Localization can be provided,
- authorization policies can be configured,
- projectin doesn't provide configuration - default iuser detail is show (redirect) only.

How to register new form

New projection can be added with the same structure in custom module. Is needed to add new content representing form projection (e.g. copy or reuse form projection from product), register new route in `routes.js`, register new route component (e.g. by generalize prepared `AbstractFormProjectionRoute`) and expose new edpoint on backend if needed (if provided `IdmIdentityProjectionDto` is insuffiscient).

Projection will obtain `entityId` route parameter with identity codeable identifier (username or uuid).

Example

Source codes are placed in example module. Product backend is reused and example provides new frontend form for new projection. Form saves identity username only.

Create new form `src/content/identity/projection/ExampleIdentityProjection.js` with content:

ExampleIdentityProjection.js

Register new route in 'routes.js':

```
{
  path: 'example/form/identity-projection/:entityId',
  component:
require('./src/content/identity/projection/ExampleIdentityProjection'),
  access: [ { type: 'HAS_ANY_AUTHORITY', authorities: ['IDENTITY_READ' ] } ]
}
```

Register new component:

```
/**
 * Example identity form projection.
 *
 * @since 10.3.0
 */
@Component(ExampleIdentityFormProjectionRoute.PROJECTION_NAME)
public class ExampleIdentityFormProjectionRoute extends
AbstractFormProjectionRoute<IdmIdentity> {

    public static final String PROJECTION_NAME = "/example/form/identity-
projection";
```

```
    @Override
    public String getName() {
        return PROJECTION_NAME;
    }
}
```

Localization

Two projections are localized by default in product. If projection with code `identity-externe` or `identity-internal` will be configured, then localization will be used.

```
...
  "eav": {
    "form-projection": {
      "identity-externe": {
        "label": "Externe user",
        "help": "Create externe user",
        "icon": "fa:walking",
        "level": "primary"
      },
      "identity-internal": {
        "label": "Internal employee",
        "help": "Create internal user",
        "icon": "fa:user",
        "level": "success"
      }
    }
  },
...
```

Localization can be added to newly configured projection in custom module. Projection localization is based on the same convetions as localization for [form definition].

Supported properties:

- **label** - projection name. Used in select boxes and as label for button to create user with projection usage.
- **help** - Used as title (tooltip) for button to create user with projection usage.
- **icon** - Used in select boxes and as icon for button to create user with projection usage (optional, `fa:user-plus` will be used as default).
- **level** - Used as level for button to create user with projection usage (optional, `default` will be used as default).

Future improvement

- Projection could be implemented by graphgl usage on backend (~extendable dto resource).
- Save button can be shown, if any section can be edited - update identity permission is needed

now.
- Assigned role attributes cannot be defined, when identity is created - add new assigned roles face mode (support projection properties).

Admin guide

- User type

Admin tutorials

- Configure and use new identity projection