FE Form & Validation

Input Field Architecture 🔗

Elements involved:

- `FormFieldStatusIndicator` which shows warning symbols and similar
- `GeneralFormFieldWrapper` which wraps the form field to integrate with the `react-form-hooks` which takes care of the form state, validation (using `yup`) and reactivity
- `InputField` which is the concrete form field

To avoid duplication within the validation rules, note how they are divided by business context (e.g. one file for each payload type) with an additional generic layer (with subtypes used at multiple places, e.g. `yupStringOptional`).

Controlled Form Fields 🔗

Mui expects the form fields to never be `undefined`, or always be `undefined`. If this is violated (e.g. an `undefined` field is written to), then mui throws an error in the console.

When editing an entity, it is however plausible that some fields are undefined. To avoid boilerplate code that sets each field to `null` prior to setting the form state, `Controlled` fields ensure values given downwards (to mui) are always "controlled" (`null` or the real value). In pseudo code, controlled form fields are `<MuiComponent value={value ? value : null} />`

Further, controlled form fields ensure only valid compounds, or `null` are exposed upwards (e.g. either a `QuarterYearDto` with both `year` and `quarter` is exposed, or `null`; but not e.g. `{ year: null, quarter: null }`). This avoid boilerplate code which cleans up the state exposed by the form on submit (e.g. remove these compounds with all `null` values).

`FormErrors` 🔗

Shows a summary of the errors occurred in the form; depending on the field including line numbers and even navigation to the erroneous field.

It needs to translate the path to the erroneous field into a label; for which a translation function has be overridden. Notably, this may be tricky with composite fields as the translation label is at a different path than the error path (e.g. `insuranceValue.amount` might has a validation error, but the translation label is `entities.stratusObject.insuranceValue`, hence without the amount suffix).