

# Take-Home Assignment: KYC Flow

## Background

You are building a KYC (Know Your Customer) flow for a global investment app. Each country has its own KYC requirements defined in a country-specific configuration file.

Your task is to create a dynamic form system that:

- Loads the KYC form configuration for a selected country
  - Renders the form fields accordingly
  - Validates user input based on rules defined in the configuration
  - Displays validation errors inline
  - Outputs the collected form data on successful submission
- 

## Requirements

### 1. Country Selector

- Provide a way to select a country (e.g., Netherlands (NL), United States (US), Germany (DE))

### 2. Dynamic Form Rendering

- Load the KYC config file for the selected country (config files stored locally in JSON or YAML format)
- Render form fields dynamically according to the configuration
- Support field types: text, number, date
- Support validation rules defined per field:
  - Required fields
  - Regex validation with custom error messages
  - Minimum and maximum length (for text)
  - Minimum and maximum value (for numbers)

### 3. Validation

- Validate all fields on submission
- Show validation errors inline for each invalid field

### 4. Submission

- On successful validation, output the collected form data (e.g., JSON printout or summary screen)

---

## Sample Config (YAML)

None

```
country: NL
fields:

- id: first_name
  label: First Name
```

```
    type: text
    required: true

- id: last_name
  label: Last Name
  type: text
  required: true

- id: bsn
  label: BSN
  type: text
  required: true
  validation:
    regex: '^\\d{9}$'
    message: 'BSN must be 9 digits'

- id: birth_date
  label: Birth Date
  type: date
  required: true

---

country: DE
fields:

- id: first_name
  label: Vorname
  type: text
  required: true

- id: last_name
  label: Nachname
  type: text
  required: true
```

```
- id: steuer_id
  label: Steueridentifikationsnummer
  type: text
  required: true
  validation:
    regex: '^\\d{11}$'
    message: 'Muss 11 Ziffern haben'

- id: birth_date
  label: Geburtsdatum
  type: date
  required: true
```

## Additional Instructions

**Handles a special requirement for the Netherlands (NL):** The `first_name`, `last_name` and `birth_date` fields **must be fetched from a mocked API and shown as read-only**, rather than entered manually. In other countries, these fields are entered manually.

- This requirement was not supported in the initial design and configs **did NOT include any metadata specifying the data source**.
- You must **design and implement an approach** to handle this special NL behavior while keeping your solution clean and extensible.
- Explain in your README or comments:
  - How you handled this requirement architecturally
  - What changes (if any) you would propose to the config format to support this in the future

## Expected Output & Deliverables

Please submit the following:

### 1. Working iOS app or Swift Playground

- Country selection interface

- Dynamic form rendering based on the provided configuration
- Form input validation
- Country-specific behavior for NL (`first_name` and `last_name` fetched from mocked API and displayed read-only)
- Form submission handling (show JSON or summary of input data)

## 2. Mocked API Implementation

- Simulated response for `/api/nl-user-profile`
- Doesn't need to be connected to a real backend (hardcoded local async response is fine)

## 3. README File

- How to run your app or playground
- Description of your solution architecture
- How the NL-specific behavior was implemented
- Your recommendation for how the config format could be improved to support data source behavior more explicitly in the future

## Evaluation Criteria

- Correctness of form behavior
- Code readability and maintainability
- Handling of dynamic configuration
- Elegance of your solution to the NL special case
- Thoughtfulness of your README and explanation

