



Case Study - Python developer

This case study is part of the GymBeam interview process for an Odoo developer role. Please read carefully the instructions, and send the case study result within 72 hours.

There are 4 assignments (+2 bonus) to be done sorted from easiest to hardest.

Prerequisite:

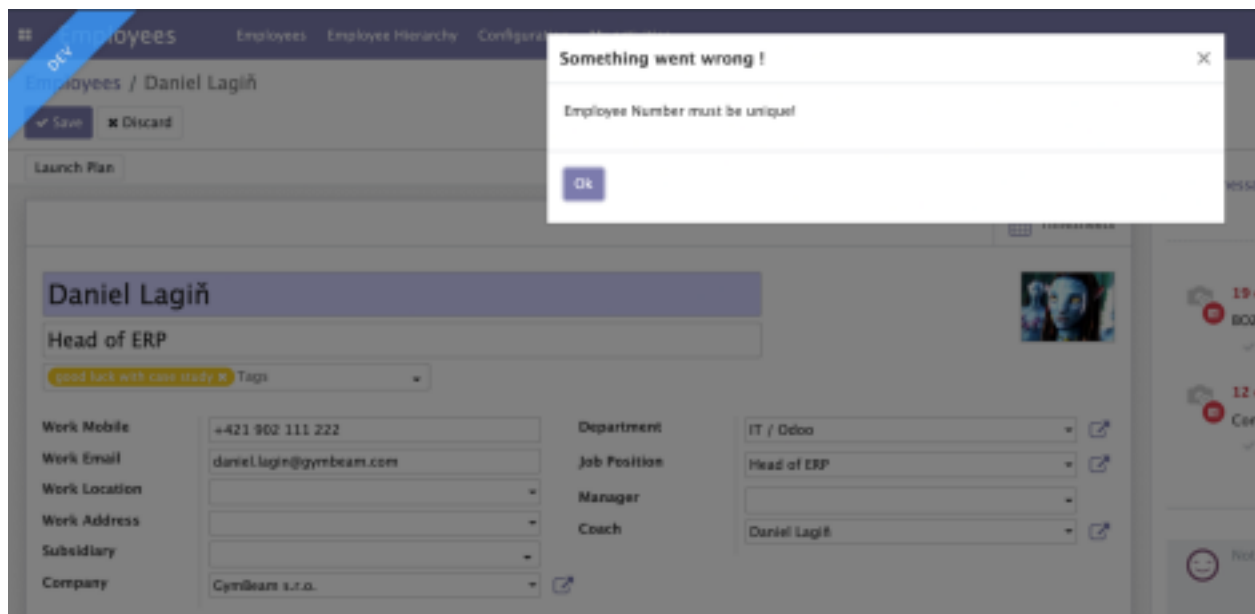
- Install Odoo 16 Community edition locally (ideally on a unix environment via docker using docker-compose / also can be done via Source Install).

Create new addon that will do following requirements:

1. I love GymBeam

1. Create a new boolean field ***i_love_gb*** on the Employee screen after the ***work_email***.
2. Create 3 new Integer fields: ***salary***, ***tax*** and ***total_salary***. Field ***total_salary*** should be automatically set to value ***salary*** + ***tax***. New fields should be after the ***address*** field on the ***Private information*** tab. Set custom text labels and translations for the added fields.
3. Replace ***phone*** field with new field ***special_phone***. On **save action** of the employee record, check if the field is empty, if it is empty, set the default value of the field to **"0901123456"**.
4. Create a new binary field ***employee_contacts*** after the new field ***i_love_gb***.
 - a. This field will be visible only if ***i_love_gb*** is set to **True**.
 - b. Create an excel file with 2+ lines. First column will be some random mail (or existing), the second one will be some random text value.
 - c. Upload this excel into a binary field.
 - d. Create a new button **Send emails** next to the **Launch plan** button. After clicking on the button Odoo will send an email with text **Welcome in GymBeam** to the email addresses in the first column. Subject of the email will be the value from the second column. **Bonus:** *Transfer this functionality to a wizard. Wizard can be opened via button click on employee form or from any place from the menu.*
 - e. As an email server, simple gmail mail can be used: [Send and Receive Emails in Odoo](#). If no correct outgoing email server is set, Odoo will store failed emails and they can be opened in GUI. Show us where.

2. Unique Employee Number



Prerequisites:

1. Create a new required & tracked char field **employee_number** on the **Employee**.
2. Create the same field on model **hr.applicant**.

User story: As a member of the data team, I need the **employee number** across the Odoo system **to be unique**.

Acceptance Criteria:

1. The **employee number** fields on both the employee and the applicant (**active** or **archived**) cannot store the same value.

3. Recruitment Website Form Validation

Prerequisites:

1. Set up online posting for Recruitment.

The screenshot shows a web form titled "Job Application Form" with a subtitle "Test". The form contains the following fields and elements:

- Your Name ***: A text input field with a red border and a red 'x' icon on the right, indicating a validation error.
- Your Email ***: A text input field with a red border and a red 'x' icon on the right, indicating a validation error.
- Your Phone Number ***: A text input field with a red border and a red 'x' icon on the right, indicating a validation error.
- Short Introduction**: A text area with a light blue border.
- Resume**: A file upload section with a "Choose file" button and the text "No file chosen".
- Submit**: A green button.
- Error Message**: A red text message at the bottom right that reads "x Please fill in the form correctly."

User story: As a **job seeker** - I want to **see error messages** when trying to **submit CV online**, so that I know the reason why my answers **were not validated**.

Acceptance Criteria:

1. Add user-friendly error messages for the application form. So after form submit the screen will look like this:

The screenshot shows the same web form after submission, with detailed error messages for each required field:

- NAME ***: A red border and a red 'x' icon. Below the field, the text "This is a required field" is displayed in red.
- EMAIL ***: A red border and a red 'x' icon. Below the field, the text "This is a required field" is displayed in red.
- PHONE NUMBER ***: A red border and a red 'x' icon. Below the field, the text "This is a required field" is displayed in red.
- SHORT INTRODUCTION**: A light blue text area.
- UPLOAD YOUR CV ***: A red border and a red 'x' icon. Below the field, the text "This is a required field" is displayed in red. The file upload section shows "Choose file" and "No file chosen".
- SEND APPLICATION ***: An orange button.
- Error Message**: A red text message at the bottom right that reads "x Please fill in the form correctly."

2. According to the screenshot above, change form **styles** and **inputs & button texts**.

Bonus: Make **Choose file input** & **No file chosen text** translatable so translations can be changed in Odoo GUI.

4. API Endpoint

Prerequisites:

1. Example JSON:

```
{
  "candidates": [
    {
      "id": 12345,
      "title": "Ing",
      "firstname": "John",
      "surname": "Doe",
      "phone": "132456789",
      "email": "applicant@odoo.com",
      "isActive": 1,
      "gender": "male",
      "job": {
        "id": 4249974,
        "title": "Job title",
        "addDate": "2021-05-11",
        "stateId": 1
      }
    }
  ]
}
```

User story: As an **HR Generalist**, I need candidates to be **created through JSON requests** so we can connect Odoo with other services & applications.

Acceptance Criteria:

1. Create a new required field ***api_id*** on the **Job Position**.
2. Create endpoint **localhost:8069/case_study/applicant/get**
3. Example JSON posted on **URL** above will:
 - Create an applicant filled with **name, phone, email & gender** parsed from JSON.
 - Link created applicant to job position based on ***api_id*** field.

Expect to spend approx. **12-16 hours** on this case study.

Choose how to deliver the result code:

1. Share a new Git repository.
2. Send us the source code files by email.

Let us know at dominika.trochanova@gymbeam.com that the case study is ready. **Good luck!**