



# Odoo assignment

Assignment version: 0.1

[https://s3-us-west-2.amazonaws.com/secure.notion-static.com/5accddc3-f855-41bd-9230-e11a39881de0/codeforward\\_Odoo\\_assignment\\_0.1.zip](https://s3-us-west-2.amazonaws.com/secure.notion-static.com/5accddc3-f855-41bd-9230-e11a39881de0/codeforward_Odoo_assignment_0.1.zip)

## Description

The goal of this assignment is to create an Odoo module that implements an ftp interface with the purpose of synchronising two systems:

- System A generates .csv files, containing information on certain devices and their contents.
- System B imports the files generated by System A, parses them and stores the information into its database.

There are two types of .csv files generated by System A:

- *devices.csv*, which contains the data on devices
- *content.csv*, which contains the content found on these devices

Examples of the files are enclosed with this assignment.

Any Odoo version can be used for this assignment.

## Models

Information needs to be stored in models that have the following definition:

### Device model

Field	Type
id	integer
name	string, up to 32 characters

Field	Type
description	string, up to 128 characters
code	string, up to 30 characters
expire date	datetime
state	string, selection: enabled, disabled or deleted

## Content model

Field	Type
id	integer
name	string, up to 100 characters
description	string, up to 128 characters
device	Many2one to the Device model
expire date	datetime
state	string, selection: enabled, disabled or deleted

Remarks:

- If expire date is in the past, the state should be assigned as 'disabled'.
- 'code' must be unique for devices.

## Requirements

1. The import job should be implemented using Odoo's cron scheduler
2. During the import, it's desirable to output every step in the log file, especially any errors that occur due to incorrectness of the data provided in the .csv files. This includes any structural errors in the .csv like missing columns etc.
3. The module should contain views to display the imported device and contents data
4. There should be a settings form where the following could be set:
  - a. Local path to the directory with the .csv files
  - b. Delimiter to use in the .csv files (by default it should be set to a comma)

## Bonus requirements

Bonus points if the following requirements are implemented as well:

1. Include unit tests to cover at least a 'happy flow' scenario
2. The solution being able to also retrieve the .csv files via FTP or SFTP instead of just locally
3. Delivering the assignment containerized/using Docker