odoo

Add custom addons path

Ex:

```
--addons-path="/home/odoo/odoo-14/addons,/home/odoo/odoo-14-custom-addons"
```

Developer mode extenstion

- Firefox:

  https://addons.mozilla.org/en-US/firefox/addon/odoo-debug/
- Chrome: https://chrome.google.com/webstore/detail/odoo-debug/hmdmhilocobgohohpdpolmibjklfgkbi?hl=en

Installing and upgrading local add-on modules

What is an Odoo add-on module?

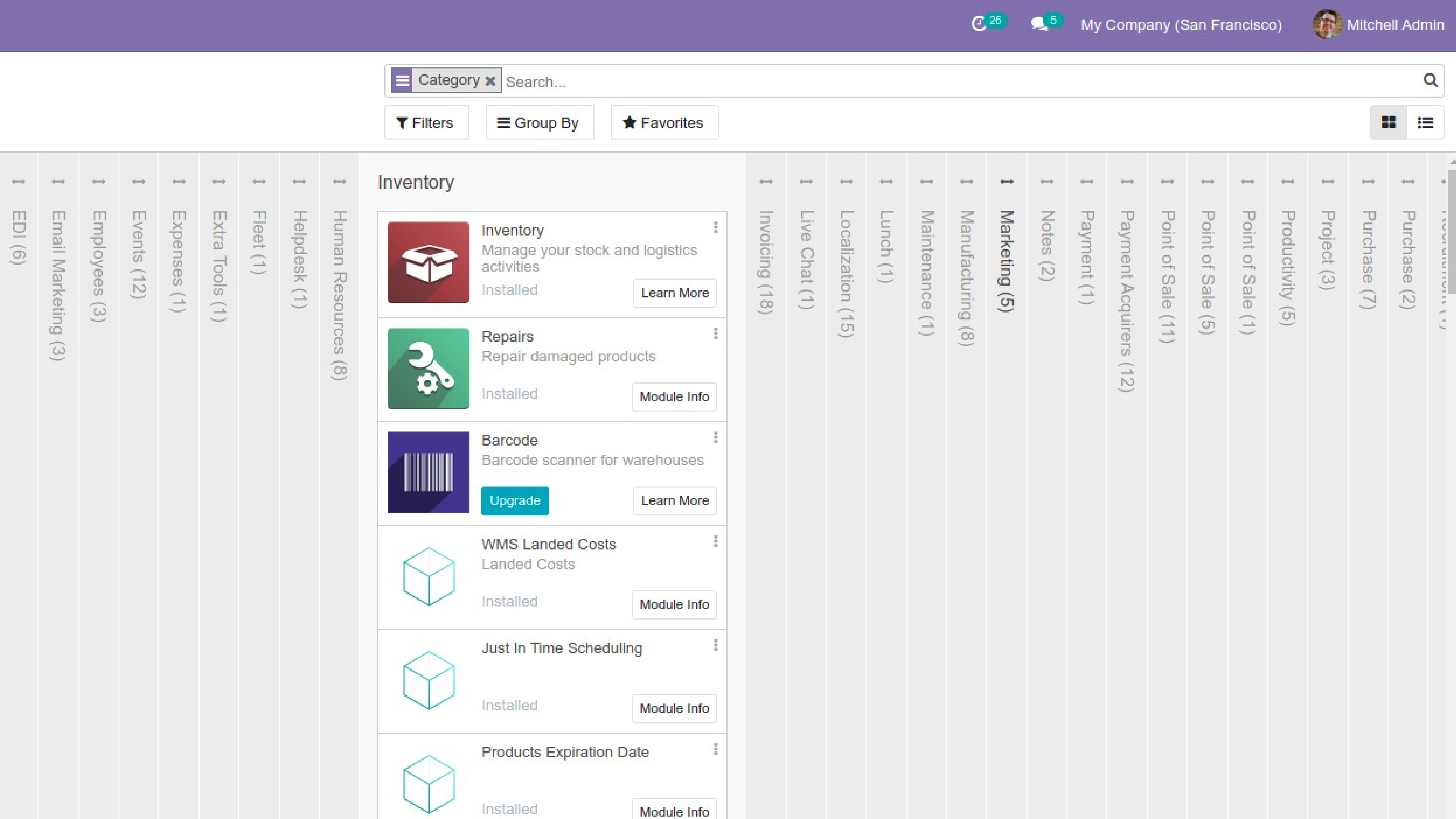
- All Odoo code are packed in the form of modules.
- These modules can be installed or uninstalled at any time from the database.
- There are two main purposes for these modules.

  1- Either you can add new apps/business logic

  2- Modify an existing application.

What is an Odoo add-on module?

- Odoo splits the features of the application into different modules. These modules can be loaded in the database on demand.
- Basically, the user can enable/disable these features at any time.
- Consequently, the same software can be adjusted for different requirements.



Inventory

Inventory
Manage your stock and logistics activities
Installed
Learn More

Repairs
Repair damaged products
Installed
Module Info

Barcode
Barcode scanner for warehouses
Upgrade
Learn More

WMS Landed Costs
Landed Costs
Installed
Module Info

Just In Time Scheduling
Installed
Module Info

Products Expiration Date
Installed
Module Info

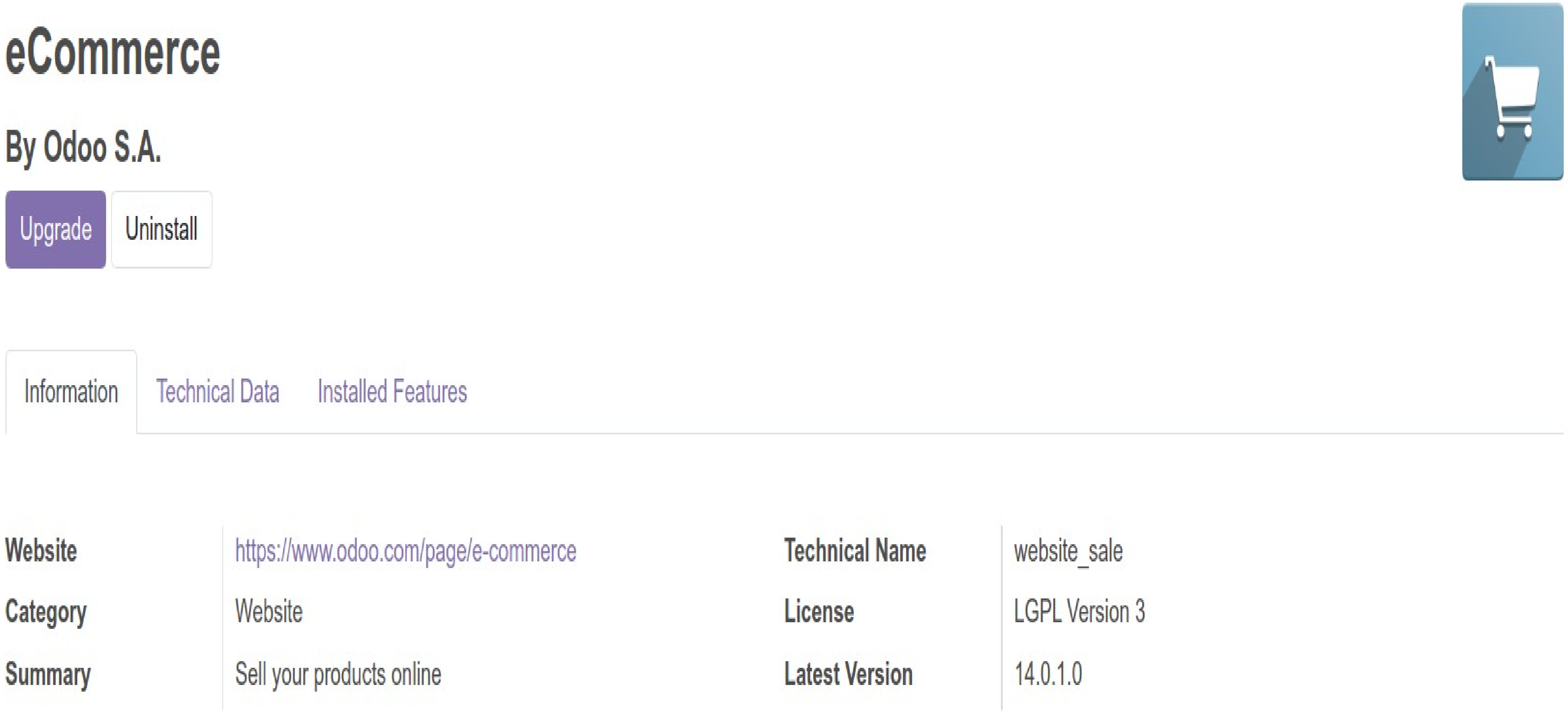
Using the scaffold command to create a module

```
$ ~/odoo-dev/odoo/odoo-bin scaffold
my_module {custom-addons-path}
```

```
$ tree my_module
my_module/
├── __init__.py
├── __manifest__.py
├── controllers
│   ├── __init__.py
│   └── controllers.py
├── demo
│   └── demo.xml
├── models
│   ├── __init__.py
│   └── models.py
├── security
│   └── ir.model.access.csv
└── views
    ├── templates.xml
    └── views.xml
```

How it works...

- An Odoo module is a directory that contains code files and other assets.
- The directory name that's used is the module's technical name.
- The name key in the module manifest is its title.



eCommerce

By Odoo S.A.

Website	https://www.odoo.com/page/e-commerce	**Technical Name**	website_sale
Category	Website	**License**	LGPL Version 3
Summary	Sell your products online	**Latest Version**	14.0.1.0

How it works...

- The __manifest__.py file is the module manifest.
- This contains a Python dictionary with module metadata including category, version, the modules it depends on, and a list of the data files that it will load.

How it works...

- The module directory must be Python-importable, so it also needs to have an __init__.py file, even if it's empty.
- This will cause the code in the __init__.py file to be executed, so it works as an entry point to run the module Python code.

How it works...

- models/ contains the backend code files
- One file per model is recommended with the same name as the model.
- for example, library_book.py for the library.book model.

How it works...

- views/ contains the XML files for the user interface, with the actions, forms,lists, and so on.
- Like models, it is advised to have one file per model.

How it works...

- data/ contains other data files with the module's initial data.
- demo/ contains data files with demonstration data, which is useful for tests,training, or module evaluation.

How it works...

security/ contains the data files that define access control lists, which is usually a ir.model.access.csv file

How it works...

controllers/ contains the code files for the website controllers that render some data into a website

How it works...

- static/ is where all web assets are expected to be placed.
- This directory mostly contains files such as JavaScript, style sheets, and images.
- They don't need to be mentioned in the module manifest .

Adding models

- In our example, we want to manage movies for a movie shop.
- To do this, we need to create a model to represent movies. =>
- Settings | Technical | Database Structure | Models

XML

- XML (Extensible Markup Language) is a markup language similar to HTML, but without predefined tags to use.
- you define your own tags designed specifically for your needs. This is a powerful way to store data in a format that can be stored, searched, and shared.

XML

- All data of the Odoo programs are stored as objects.
- Views are defined to expose these objects to the user.
- Odoo uses a dynamic user interface, which means it is not statically built by some codes, it is dynamically built from XML descriptions. And these screen descriptions are called views.

Adding menu items and views

- To add a view, we will add an XML file with its definition to the module.
- Since it is a new model, we must also add a menu option for the user to be able to access it.=>

Adding access security

Defining the model representation and order

- _rec_name is used to set the field that's used as a representation or title for the records.
- The other one is _order , which is used to set the order in which the records are presented.

Adding data fields to a model

- Char is used for string values.
- Text is used for multiline string values.
- Selection is used for selection lists. This has a list of values and description pairs.

  The value that is selected is what gets stored in the database, and it can be a string or an integer.

Adding data fields to a model

- Html is similar to the text field, but is expected to store rich text in an HTML format.
- Binary fields store binary files, such as images or documents.
- Boolean stores True/False values.

Adding data fields to a model

- Date stores date values.You can use fields.Date.today() to set the current date as a default value in the date field.
- Datetime is used for datetime values. They are stored in the database in UTC time.

Adding data fields to a model

- The Integer fields.
- The Float fields store numeric values. Their precision can optionally be defined with a total number of digits and decimal digit pairs.
- Monetary can store an amount in a certain currency.

Field Optional Attributes

string is the field's title, and is used in UI view labels. It is optional. If not set, a label will be derived from the field name by adding a title case and replacing the underscores with spaces.

Field Optional Attributes

default is the default value. It can also be a function that is used to calculate the default value;

for example, default=_compute_default , where _compute_

default is a method that was defined on the model before the field definition.

Field Optional Attributes

- The readonly flag makes the field read-only by default in the user interface.
- The required flag makes the field mandatory by default in the user interface.
- Other attributes will be discussed later.

Adding relational fields to a model

Relations between Odoo models are represented by relational fields. There are three different types of relations:

1-many-to-one,

2-one-to-many,

3-Many-to-many =>