Chapter 5 Basic Server Side Development
Short Note

Key points of these chapter is provided here:

- While writing new methods if you don't use decorator, then the method is executed on a recordset. 'self' property of a model is a recordset that can refer to an arbitrary number of database records (this includes empty recordsets), and the code will often loop over the records in self to do something on each individual record.

Note on Recordset: Record means any single record (single row of data) and recordset means number of records (multiple row of data). it's like data row in database table. in the term of odoo, like a create method return single record.

- The **@api.model** decorator works like Python's **@classmethod** decorator. Method name and button name must be same. This will load all the functionality of button allinged. Upon clicking this button, the Odoo web client will invoke the Python function mentioned in the name attribute.

- If we don’t want to allow some actions for button call we can raise error exception. When an exception is raised in Python, it propagates up the call stack until it is processed. In Odoo, the **RPC** (remote procedure call) layer that answers the calls made by the web client catches all exceptions and, depending on the exception class, triggers different possible behaviors on the web client. Any exception not defined in **odoo.exceptions** will be handled as an internal server error (HTTP status 500) with the stack trace. In all cases, the current database transaction is rolled back. There are some exception classes defined in **odoo.exceptions**, all deriving from the base legacy **except_orm** exception class.

- A function with a strange name, **_()** , which is defined in **odoo.tools.translate** . This function is used to mark a string as translatable, and to retrieve the translated string at runtime, given the language of the end user that's found in the execution context. When using the _() function, ensure that you pass only strings with the interpolation placeholder, not the whole interpolated string. For example, **_('Warning: could not find %s') % value** is correct, but **_('Warning: could not find %s' % value)** is incorrect because the first one will not find the string with the substituted value in the translation database.

- Odoo will catch this error and display a traceback to the user. If you don't want to show a full error log to the user, you can cache the error and raise a custom exception with a meaningful message.

- At startup, Odoo loads all the modules and combines the various classes that derive from Model , and also defines or extends the given model. These classes are stored in the Odoo registry, indexed by name. The env attribute of any

recordset, available as **self.env**, is an instance of the Environment class defined in the **odoo.api** module. If you know the name of the model you're looking for, **self.env[model_name]** will get you an empty recordset for that model. Environment class has a **cr** attribute, which is a database cursor can be used to pass raw SQL queries. It has a **user** attribute, which is a reference to the current user performing the call. It has a context attribute, which is a dictionary that contains the context of the call. This includes information about the language of the user, the time zone, the current selection of records, and much more.

- Any method working with one record must starts by checking whether the book recordset that's passed as self contains exactly one record by calling **ensure_one()**.

- **write()** method works for recordsets of arbitrary size and will update all records with the specified values in one single database operation when the two previous options perform one database call per record and per field. It does not work if the records are not yet present in the database.

- **search()** method takes domain as input. The method returns a recordset that contains all the records that matches the domain. The security rules ensure that the user only gets those records to which they have read access rights. Following keyword arguments are also supported:

```
domain = [
    '|',
        '&', ('name', 'ilike', 'Book Name'),
             ('category_id.name', '=', 'Category Name'),
        '&', ('name', 'ilike', 'Book Name 2'),
             ('category_id.name', '=', 'Category Name 2')
]
books = self.search(domain)
```

Example of search domain in codebase

• **offset=N :** This is used to skip the first N records that match the query. This can be used along with limit to implement pagination or to reduce memory consumption when processing a very large number of records. It defaults to 0 .
• **limit=N :** This indicates that, at most, N records should be returned. By default, there is no limit.
• **order=sort_specification :** This is used to force the order in the recordset returned. By default, the order is given by the _order attribute of the model class.
• **count=boolean :** If True , this returns the number of records instead of the recordset. It defaults to False .