Odoo Snippets

https://marketplace.visualstudio.com/items?itemName=jigar-patel.OdooSnippets

This extension contains predefined code snippets for Odoo developer. Snippets are available for following supported languages (file extensions):

- Python (`.py`)
- Javascript (`.js`)
- XML (`.xml`)
- CSV (`.csv`)

All snippets follow the Odoo Guidelines.

Installation

Launch VS Code. Press `Ctrl+P` or `Command+P` (for Mac). Paste command `ext install jigar-patel.OdooSnippets` and press `Enter`.

How to use?

Snippets prefixes are defined in documentation. Type prefix and select snippet from the suggestion list.

For example if you want to add `create` CRUD method then type `oocreate`.

Search Snippets

You can search snippets with `Odoo Snippet Search` command which can be run by pressing `Ctrl+Shift+P` or `Command+Shift+P` (for Mac).

Python (`.py`)

Prefix	Purpose
`manifest`	Manifest JSON
`imo`	Import Odoo Model/Fields/API
`ime`	Import Odoo Exceptions
`iml`	Import Logging
`imf`	Import From
`oomodel`	Add model
`oofchar`	Char Field
`ooftext`	Text Field
`oofhtml`	Html Field
`oofboolean`	Boolean Field
`oofinteger`	Integer Field
`ooffloat`	Float Field
`oofselection`	Selection Field
`oofdate`	Date Field
`oofdatetime`	DateTime Field
`oofimage`	Image Field
`oofbinary`	Binary Field
`oofmonetary`	Monetary Field
`oofmany2one`	Many2one Field
`oofone2many`	One2many Field
`oofmany2many`	Many2many Field
`oofcompute`	Compute Field
`oofcomputei`	Compute Field with Inverse
`oofonchange`	Onchange Field
`oofconstrains`	Constrains
`ooone`	Ensure One
`oooverride`	Override Method
`oocreate`	Create Method
`oowrite`	Write Method
`oounlink`	Unlink Method
`related`	Related Attribute
`required`	Required Attribute
`readonly`	Readonly Attribute
`copy`	Copy Attribute
`store`	Store Attribute
`digits`	Digits Attribute
`Help`	Help Attribute
`check_company`	Check Company Attribute

XML (`.xml`)

Prefix	Purpose
`<odoo>`	Odoo Namespace
`<data>`	Data Tag
`<menuitem>`	Root Menu
`<menuitem>`	Submenu
`<form>`	Form View
`<tree>`	Tree View
`<kanban>`	Kanban View
`<search>`	Search View
`<calendar>`	Calendar View
`<cohort>`	Cohort View
`<gantt>`	Gantt View
`<graph>`	Graph View
`<pivot>`	Pivot View
`<activity>`	Activity View
`<qweb>`	Qweb View
`<viewInherit>`	Inherit View
`<buttonHeader>`	Button in Form View Header
`<buttonBox>`	Stat Button Container
`<buttonStat>`	Stat Button
`<field>`	Field Tag
`<filter>`	Filter in Search View
`<notebook>`	Notebook
`<page>`	Notebook Page
`<chatter>`	Chatter
`<action>`	Action
`<actionView>`	View in Action
`<actionServer>`	Server Action
`<actionUrl>`	URL action
`<actionClient>`	Client Action
`<actionReport>`	Report Action
`<report>`	Report Template
`<cron>`	Cron
`<record>`	Record
`<function>`	Function
`<template>`	Template
`<templateInherit>`	Inherit Template
`<xpath>`	Xpath
`<xpathAttr>`	Xpath to Attributes
`<xpathField>`	Xpath for Field
`<attribute>`	Attribute for Xpath
`<group>`	Group
`<script>`	Script
`<link>`	SCSS Link
`<assetsBackend>`	Backend Assets
`<assetsFrontend>`	Frontend Assets

Qweb

Prefix	Purpose
`<t-if>`	If Condition
`<t-if-else>`	If Else Condition
`<t-foreach>`	Foreach Loop
`<t-set>`	Set Variable
`<t-esc>`	Evaluate Expression
`<t-out>`	Evaluate Expression
`<t-raw>`	Evaluate Html Content
`<t-call>`	Call Other Template
`<t-debug>`	Invokes a Debugger
`<t-log>`	Log the Expression Result
`t-att`	t-att
`t-attf`	t-attf

Javascript (`.js`)

Prefix	Purpose
`odoo-module`	Define Odoo Module
`imc`	Import Component
`imcs`	Import Component and useState
`oc`	Component
`occ`	Component with Constructor
`ocs`	Component with Setup
`ms`	Setup
`mws`	willStart
`mm`	mounted
`mwup`	willUpdateProps
`mwp`	willPatch
`mp`	patched
`mwum`	willUnmount
`mce`	catchError
`uss`	useState
`usr`	useRef
`usc`	useContext
`ust`	useStore
`usd`	useDispatch
`onm`	onMounted
`onwn`	onWillUnmount
`onwp`	onWillPatch
`onp`	onPatched
`ows`	onWillStart
`owup`	onWillUpdateProps

Full Expansions

`odoo-module` - Define Odoo Module

```
/** @odoo-module **/
```

`imc` - Import Component

```
const { Component } = owl;
```

`imcs` - Import Component and useState

```
const { Component } = owl;
const { useState } = owl.hooks;
```

`oc` - Component

```
export class MyComponent extends
Component { }
```

`occ` - Component with Constructor

```
export class MyComponent extends
Component {
    constructor(...args) {
        super(...args);
    }
}
```

`ocs` - Component with Setup

```
export class MyComponent extends
Component {
    setup() {
        super.setup();
    }
}
```

`ms` - Setup

```
setup() {
    super.setup();
}
```

`mws` - willStart

```
async willStart() { }
```

`mm` - mounted

```
mounted() { }
```

`mwup` - willUpdateProps

```
async willUpdateProps(nextProps) { }
```

`mwp` - willPatch

```
willPatch() { }
```

`mp` - patched

```
patched() { }
```

`mwum` - willUnmount

```
willUnmount() { }
```

`mce` - catchError

```
catchError(error) { }
```

`uss` - useState

```
this.state = useState({});
```

`usr` - useRef

```
this.ref = useRef('');
```

`usc` - useContext

```
this.context = useContext();
```

`ust` - useStore

```
this.state = useStore((state) =>
state);
```

`usd` - useDispatch

```
this.dispatch = useDispatch();
```

`onm` - onMounted

```
onMounted();
```

`onwu` - onWillUnmount

```
onWillUnmount();
```

`onwp` - onWillPatch

```
onWillPatch();
```

`onp` - onPatched

```
onPatched();
```

`onws` - onWillStart

```
onWillStart();
```

`onwup` - onWillUpdateProps

```
onWillUpdateProps(nextProps => { });
```

Contributing

If you'd like to report an issue or help contribute to this visual studio code extension, please visit the repository on Github.

- Repository
- Issues