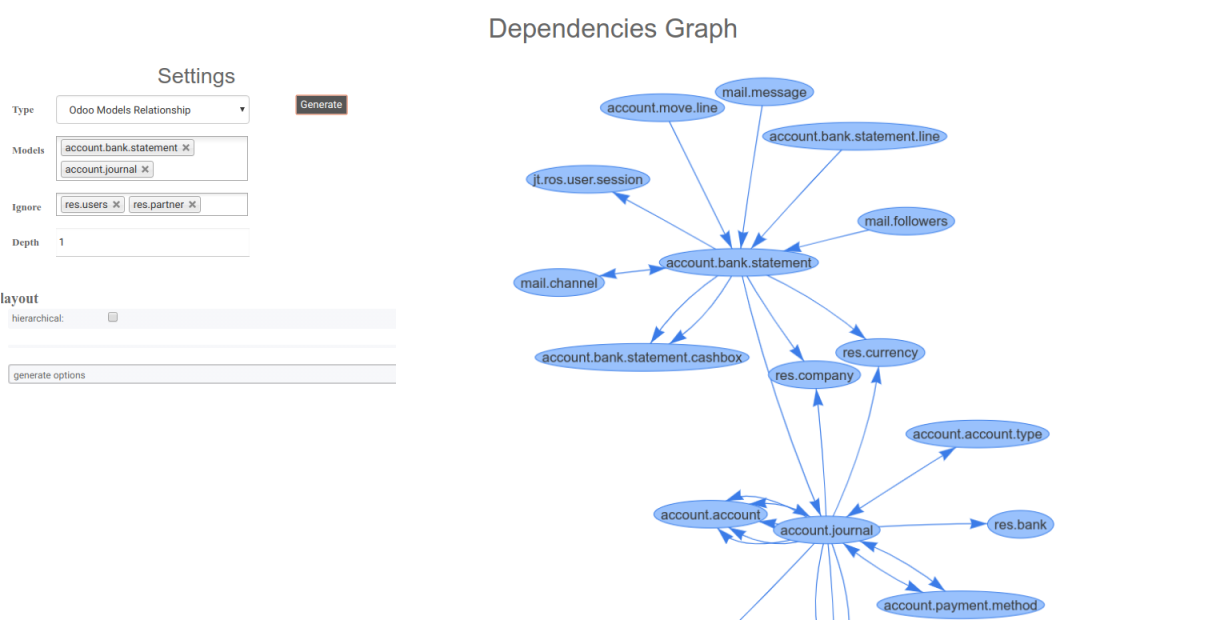
Odoo Dependencies

Odoo Dependencies Graph



This tool is aimed to help software developers to get a better understanding of the dependencies among the modules and the models relationship in an Odoo application. It also shows the inheritance graph of JavaScript objects defined in Odoo.

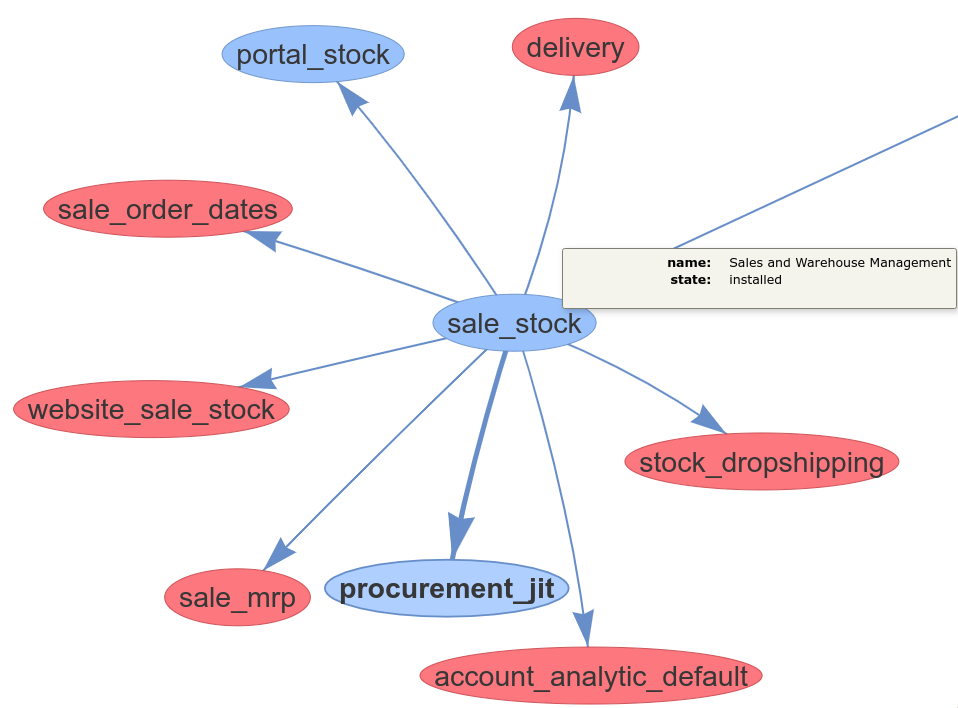
The library vis.js is used to render the graph.

Modules Dependencies

**Odoo Module Parents:** Given an Odoo module it shows every module in which the module depends on. These are the modules that need to be installed before it.

**Odoo Module Children:** Given an Odoo module it shows every module that depends of it directly or indirectly.

A module can have redundant dependencies. The option **Acyclic graph** shows or hides these redundant dependencies.

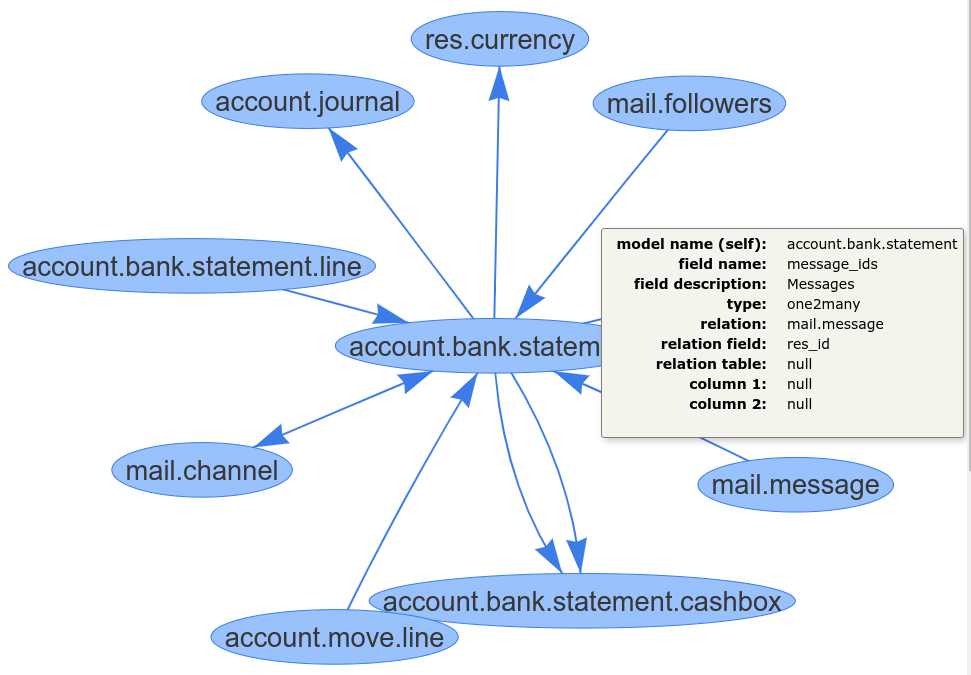


Models Relationship

Given some models it shows the relationship graph with other models.

Some models, for example **res.users**, have relations with all models. The option **Ignore** exclude some models from the graph.

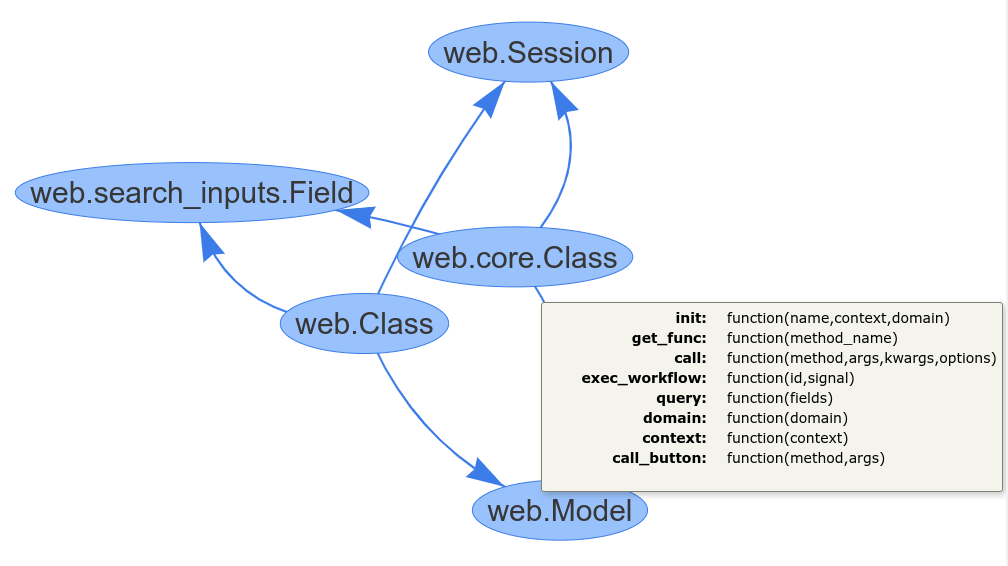
The option **Depth** specifies how deep the graph is going to be. A lower value is recommended.



## JS Objects Inheritance

**JavaScript Parents:** Given some JavaScript constructor functions, the graph shows the functions (parents) which the given functions extend.

**JavaScript Children:** Given some JavaScript constructor functions, the graph shows the functions (children) that extend the given functions.



# **External Dependencies ,Pre Init Hook & Post Init Hook In Odoo**

When we creating a module in odoo important thing is \_\_manifest\_\_.py( for odoo 10.0 and later versions) or \_\_openerp\_\_.py(for odoo 9.0 and earlier versions).

In manifest file, we can define

1. **name (str, required):-** Human readable name of the module(other than the technical name)
2. **version*(*str*)* :-** Module version(should follow version rules).
3. **summary*(*str*):-***Shortdescription of the module.
4. **description (str):-**Extended description for the module.
5. **author (str):-**Module author’s name.
6. **website (str):-** Module author’s website URL.

**license (str, defaults: AGPL-3):-** Distribution license for the module.

1. **category *(str, default: Uncategorized)*:-** Classification category within Odoo.
2. **depends (list(str)):-** Depended on Odoo modules which must be loaded before the module.
3. **data (list(str)):-**Path of all data files from the root directory which must always be installed or updated with the module.
4. **demo (list(str)):-** Path of all demo data files which are only installed or updated in demonstration mode.
5. **auto\_install (bool, default: False):-**If True, the module will automatically be installed if all dependent modules are installed.

Except for those keys in odoo \_\_manifest\_\_.py we have some special keys for module compatibility check, adding required dependencies and for running scripts during module installation. Those keys are

-pre\_init\_hook

-External\_dependencies

-post \_init\_hook

**External dependency**

In \_\_manifest\_\_.py file extenal\_dependency key is used for declaring all external libraries (python packages or any binaries) that have to be installed to make the module work. External dependencies are defined in \_\_manifest\_\_.py file under external dependency key like below

"*external\_dependencies": {"python": [], "bin": []},*

Check if some Python library exists:

"external\_dependencies": {"python" : ["openid"]}

Check if some system application exists:

"external\_dependencies": {"bin" : ["libreoffice"]}

If someone created a module in Odoo, which uses Python modules for its working. So at the time of module installation if the system does not find Python packages or any binaries defined in \_\_maifest\_\_.py installed in the system, it will raise an error and not allow to install the module.

**For example:**

In Odoo's  ‘Weighing Scale Hardware Driver’ module’s \_\_manifest\_\_.py file there is a line

*‘external\_dependencies’: {‘python’:[‘serial’]}*

For defining python dependency file ‘scale’ for the module ‘Weighing Scale Hardware Driver’, and this module allows the point of sale to connect to a scale using a USB HSM serial scale interface such as Mettler Toledo Ariva. The module requires Python dependency of ‘scale’ library, so in the module they use external\_dependencies’.So at the time of module installation, if the system does not find ‘scale’ library installed, it will raise an error and not allow to install the module.

**Pre Init Hook:**

The ‘pre\_init\_hook‘ is also a special key available in  \_\_manifest\_\_.py. During installation, when the user clicks to install the custom module, it will call a Pre-installation hook and do some initialization, copying of files or other and then after that once it finished, we can capture the return installation process of Odoo and run a Post-installation hook.

The pre\_init\_hook is called with a database cursor and may perform modifications in the database to prepare for the module installation.

You can use pre\_init\_hook’ before registering the module's logic in the ir.module.module. It is commonly used for modules compatibility check.

For using the pre\_init\_hook define this key in your \_\_manifest\_\_.py and assigning it’s a method created in your \_\_init\_.py of the module(Value of the key is the name of a Python function which must be defined in \_\_init\_\_.py).

***For Example :***

As many times we also have validated that the module build for Odoo 8 should not be installed in Odoo 9 in that case we use pre\_init\_hook .

In  \_\_opnerp\_\_.py:‘pre\_init\_hook’:’version\_check’In  \_\_init\_\_.py:def **version\_check**(cr):    # version check     return True

**Post Init Hook:**

The post init hook is called with a database cursor and registry object and may perform modifications in the database to finalize the module installation. We can define the ‘post\_init\_hook‘ in the same ways as pre\_init\_hook.

***For Example :***

In odoo Invoicing module \_\_manifest\_\_.py they using post init hook

For auto installing l10n module

In  \_\_openerp\_\_.py:‘Post\_init\_hook’:’\_auto\_install\_l10n’,

In \_\_init\_\_.py:def **\_auto\_install\_l10n**(cr, registry):    #check the country of the main company (only) and eventually load some module needed in   that country    #auto install localization module(s) if available

References

1. <https://apps.odoo.com/apps/modules/10.0/dependencies_graph/>

2. <https://www.cybrosys.com/blog/pre-post-hooks-odoo>