

Table of Contents

<i>Setting Up Your Computer</i>	3
Before Getting Ready For Development	3
Stuff To Know	3
General Setup	3
Browser Tools	3
Shell	4
Editors	5
Organizing Projects	5
Nice To Have	6
Mac	6
Linux	6
Ready To Go	6

Setting Up Your Computer

Before Getting Ready For Development

Before reading through this wiki, make sure you've gone through the [general wiki information on the home page](#) as well as [our information about company tools](#). This will be helpful to get a rough idea of everything that we are using as a company before you jump into Odoo specific or development specific tools.

Stuff To Know

Every operating system is going to be slightly different, but we are going to focus primarily on Mac OS and Ubuntu (Linux) systems. There is not too much that you will need to run ERP projects for Blue Stingray because we try to do all of our local development through Docker containers and every project should come with a docker-compose file.

It will be helpful to read through some general documentation to get your head around the tools that we use as a team before you start working on projects or setting up your computer. You do not need to know every tool front to back, but you need to know each tools purpose:

- [Odoo](#) is the ERP system that we work with primarily. Read through the marketing materials and make sure you understand what an ERP is, why companies use them/need them, and some of the major features offered by an ERP system.
- Shell. Make sure you know your way around a shell. Look through [a cheat sheet of commands](#) or get familiar with man.
- Password Managers. Understand the point of password managers such as Last Pass.
- Version Control and Source Repositories. We use Bitbucket but understand the concepts of version control and git in general will help with using any version control system.
- SSH. Make sure you know how to setup and use SSH keys on your system.
- IDEs/Editors. You'll pick whatever you are comfortable with, but you will want to know at least one editor very well. Becoming famaliar with their plugins and extensions helps a lot too.

General Setup

Okay, once you feel comfortable with all of the topics above, here are some steps to follow for making sure your computer has everything you need for local development:

Browser Tools

There are a few things related to browsers and browser tools that you'll want to add:

- **Browser.** [Chrome](#) or [Firefox](#) The Odoo ERP system runs best in these browsers. Do not worry about

running it in IE, Safari, etc.

- **Lastpass.** Set up [LastPass](#). There is a browser extension and we will share passwords with each other internally.
- **Bitbucket.** Generate an ssh key locally with `ssh-keygen` if you don't already have one and upload that to your Bitbucket account. You can use any email for your Bitbucket account. Ask someone to add you to the Blue Stingray team on Bitbucket.

Shell

Now setup your shell/terminal and some tools that you'll need for building and running projects:

- **The Shell.** Depending on your system, you'll want to familiarize yourself with a shell. Look at customizing your `bashrc` to fit your personal development or use a tool such as [Oh My Zsh](#).
 - **Mac:** You will want to use either the default Terminal or something like [iTerm](#).
 - **Linux:** The default Terminal works well.
- **Git.** Download [Git](#).
 - **Mac:** This will be prepackaged with Developer Tools. If you try to run `git` in your terminal it will prompt you to download Developer Tools if you don't have it already.
 - **Linux:** Install with your package manager using something along the lines of `apt install git`.
 - **Bash Completion - Both:** You should enable bash completion for Git to enable tab completion for branches.
<https://github.com/bobthecow/git-flow-completion/wiki/Install-Bash-git-completion>.
- **Homebrew.** **Mac:** For Mac, install [Homebrew](#).
- **Python.** Install `python3`. You do not need `python2` on your local at least for odoo development. If you happen to be working on an older version on Odoo then `python2` will come packaged inside the container, but you will still run `python3` locally. `Pip3` should come packaged with `python3` but make sure that you have that available as well.
 - **Mac:** You can run `brew install python3`
 - **Linux:** Install with your package manager using something along the lines of `apt install python3`.
- **PIP.** After installing `python3` and `pip3`, make sure that the `pip` executable is in your `$PATH`. For Mac and Linux this will be the same. Make sure that you have a line in either your `.bashrc` or your `.bash_profile` that points to your `python3` bin. For Mac it may look like `export PATH=$PATH:/Users/{my-user}/Library/Python/3.7/bin`. **Find your bin and add it to the end of your path or your going to have some problems running the internal tools we have created.**
- **Docker.** Install `docker` and `docker-compose`. You can check if you have `docker` and if it's running properly by running `docker ps`. I recommend not running it as `sudo` if you can avoid it.
 - **Mac:** Download [Docker For Mac](#). After you download it, you will want to run it from your Applications. After running, you will small a small docker image on your top bar where you can start/stop docker.
 - **Linux:** Install with your package manager using something along the lines of `apt install docker docker-compose`. On Linux after you download make sure that your user is in the `docker` group via `sudo usermod -aG docker ${USER}`. [Further instructions](#).

Editors

See [Configuring your Editor](#)

Organizing Projects

Do whatever is best for you. Make yourself efficient.

One great way to try organizing projects, and something that will prevent code conflicts, is to designate an area for projects and make sure that you have a folder for each version that you are working on. For example, you may set up something like:

[snippet.sh](#)

```
$ ~/Work

$ ~/Work/projects # This would store our repos and active development.
$ ~/Work/documents # This would store any random downloads that we need
temporarily.
$ ~/Work/design # This may contain some design assets that we need.
```

The projects section that you could break up even more if you feel that it helps you. Some developers may put /projects/clients/... or /projects/internal/... depending on what is easiest to understand. For this example, I'll just group everything together.

So we will want to have client folders because each client may have multiple projects:

[snippet.sh](#)

```
$ ~/Work/projects/bell_racing # Bell Racing is a client that may have
multiple projects
$ ~/Work/projects/empac_group # Empac Group is a client that may have
multiple projects
$ ~/Work/projects/bhs # BHS is a client that may have multiple projects
```

Now if we take a specific client, for example, we will want to have a nested directory for each version of a project. We will look at Bell Racing. Bell Racing has an Odoo instance running on version 9. This is a git repository stored at bitbucket.org/bluestingray/bell_addons.git. They are also transitioning to a version 11 instance. So this is going to be the same repo, but a different branch.

In most development scenarios, you would just keep the code under a single directory that you cloned down but switch between branches. Since we are using docker, and because there are such large differences between versions of Odoo, you will want a separate `git clone` for each version:

snippet.sh

```
$ ~/Work/projects/bell_racing/bell_odoo_v9 # This is  
bitbucket.org/bluestingray/bell_addons.git on branch=9.0  
$ ~/Work/projects/bell_racing/bell_odoo_v11 # This is  
bitbucket.org/bluestingray/bell_addons.git on branch=11.0
```

Nice To Have

We'll keep a list of nice to have applications, for the ERP Implementation Team specifically, listed out here. We will split them out into Linux and Mac.

Mac

- Alfred (Spotlight replacement)
- Franz
- iTerm

Linux

- Terminator

Ready To Go

You should have everything that you need now. Jump back to the [Team wiki page](#) and start setting up some projects and making sure that you can clone down an existing project, run it without any issues, and that you understand the process for creating a new ERP project.

From:
<http://wiki.bluestingray.com/> - **Blue Stingray Wiki**

Permanent link:
<http://wiki.bluestingray.com/teams:erp:process:setup-environment?rev=1568918802>

Last update: **2020/12/22 23:26**

