

Web applications with Python

- Objectives
 - Interacting with Packages
 - Learn about the choices of web frameworks
 - See why Pyramid is a good general-purpose choice
 - Install Pyramid and start with a scaffolded project
 - Layout your content (CSS, JS, Templates, etc.) correctly
 - Create views and templates
 - Define URL routes to control sites URLs
 - Build dynamic HTML templates that map data to HTML

Consuming libraries [virtual environments]

- Virtual environments allow multiple Python projects that have different (and potentially conflicting) requirements, to coexist on the same computer.
- With virtual environments you can
 - Store multiple versions of a package


Consuming libraries [third-party packages]

- Python has several package managers which install and upgrade third-party packages
 - **pip** (default since 3.4)
- These are similar to
 - NPM from node.js [[1](#)]
 - NuGet from .NET [[2](#)]
 - Gems from Ruby [[3](#)]

Consuming libraries [third-party packages]

- Installing packages
 - **pip install <packagename>**

Install requests package




```
(env) C:\>pip install requests
Downloading/unpacking requests
  Downloading requests-2.0.1.tar.gz (412kB): 412kB downloaded
  Running setup.py egg_info for package requests

Installing collected packages: requests
  Running setup.py install for requests

Successfully installed requests
Cleaning up...
```

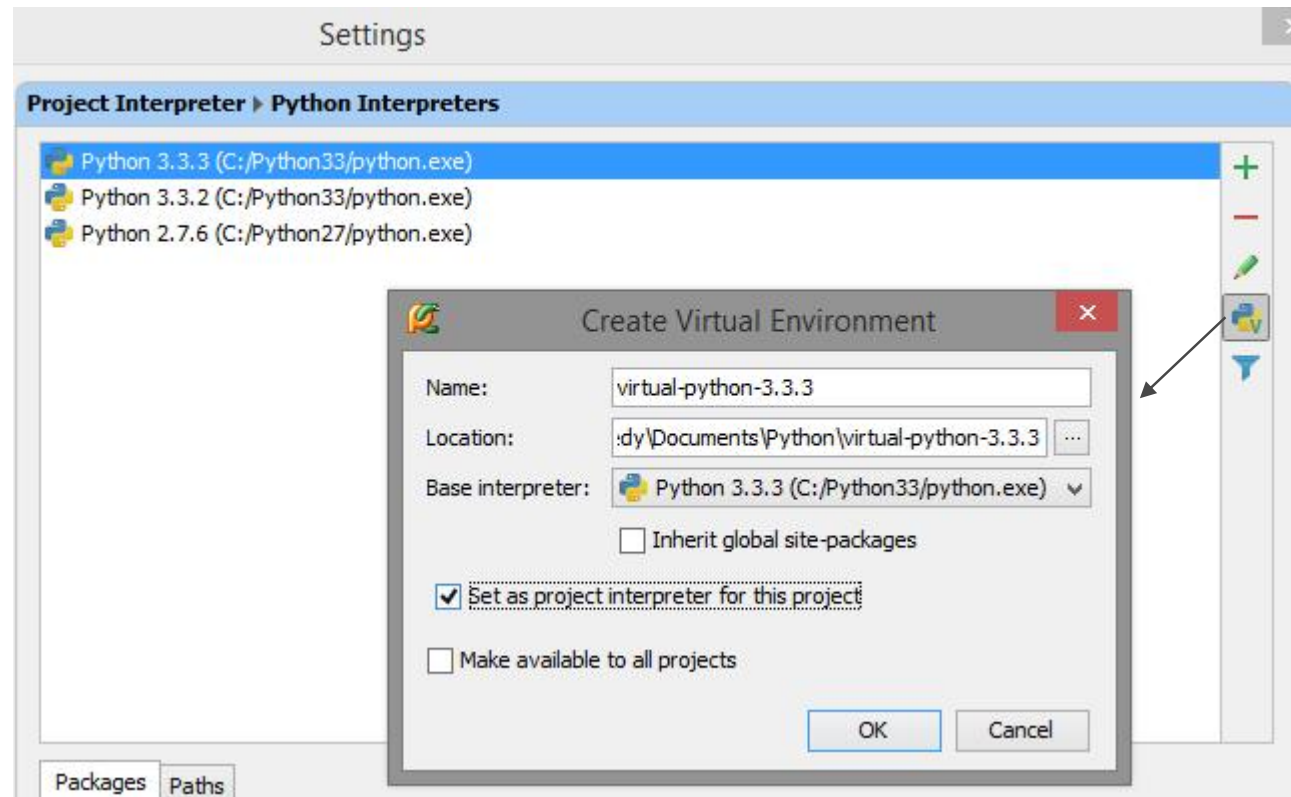
Use package



```
(env) C:\Users\Michael Kennedy\Documents\Python\SecondProj>python
>>> import requests
>>> r = requests.get("http://www.develop.com")
>>> len(r.text)
31635
>>>
```

Consuming libraries [PyCharm]

- PyCharm has support for virtual environments
 - Can isolate environment
 - Can inherit global package settings

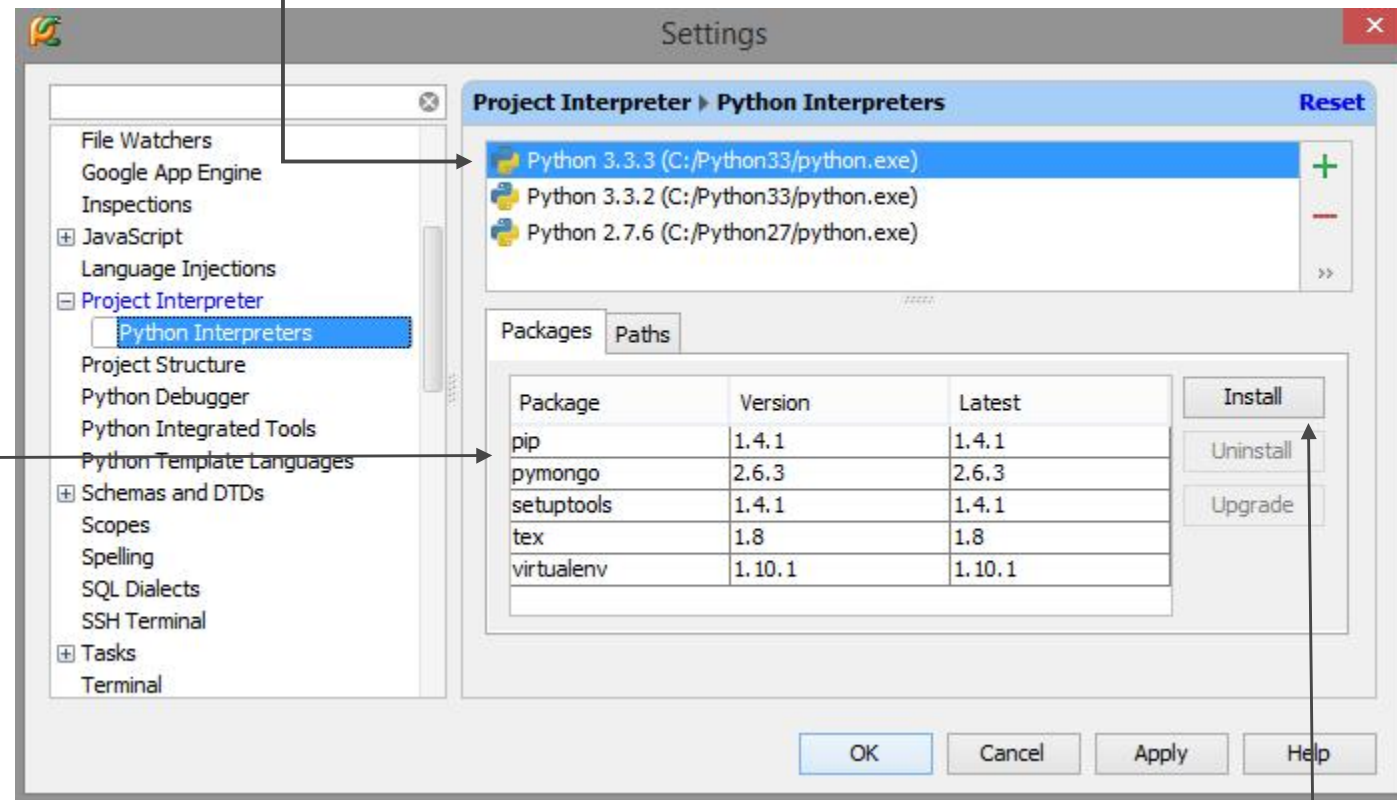


Consuming libraries [PyCharm]

- PyCharm has support for package management

For a selected interpreter or virtual environment

Manage / update installed packages



Install new packages

Pyramid Architecture



- www.pylonsproject.org
- micro framework
- arbitrary web application
- RESTful request dispatching

Sites using Pyramid

- Dropbox
- Reddit
- Digg
- O'Reilly
- Source Forge

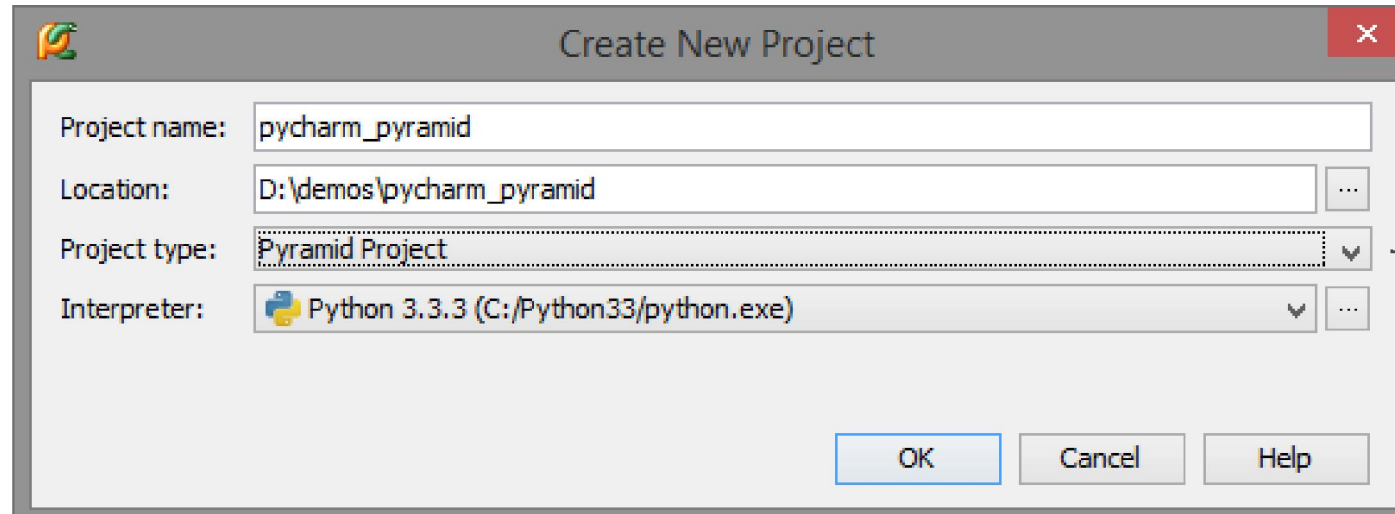


Building blocks of Pyramid

Routing	Views	Static files
Models	View Classes	Templates
Servers	Configuration	Deployment
Scaffolding		

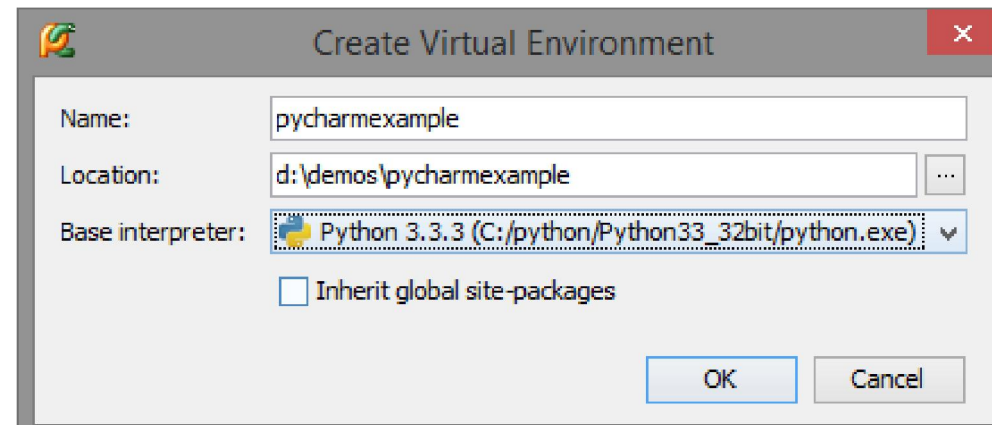
Getting started [Installing Pyramid via PyCharm]

1.



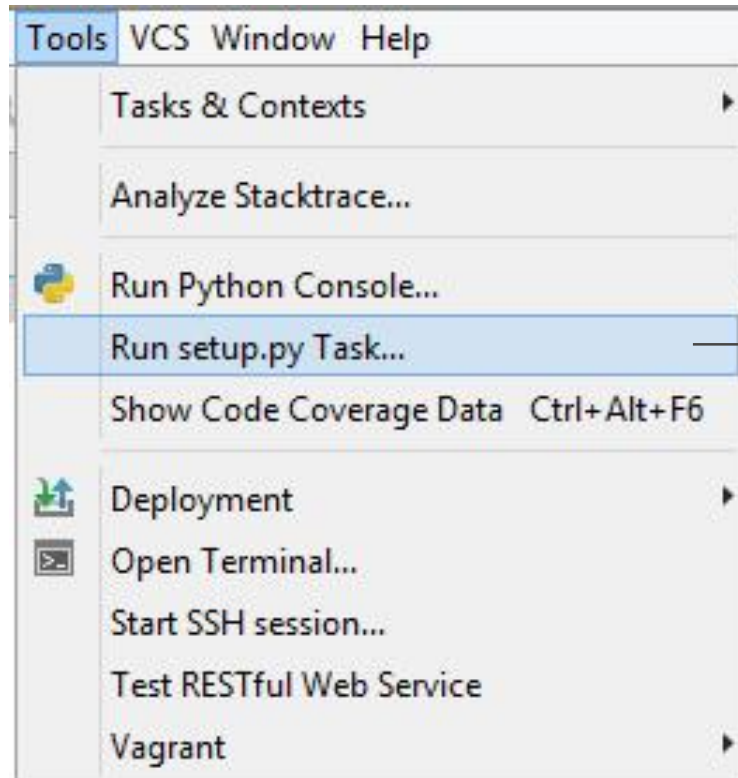
Choose File > New Project > Pyramid Project

2.



Add a virtual environment via PyCharm's tools

Getting started [Installing Pyramid via PyCharm]



3.

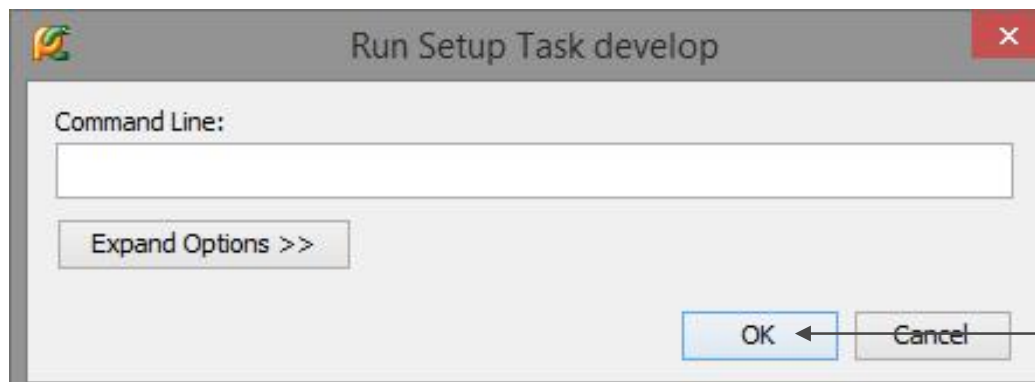
4. Choose develop

Enter setup.py task name

develop

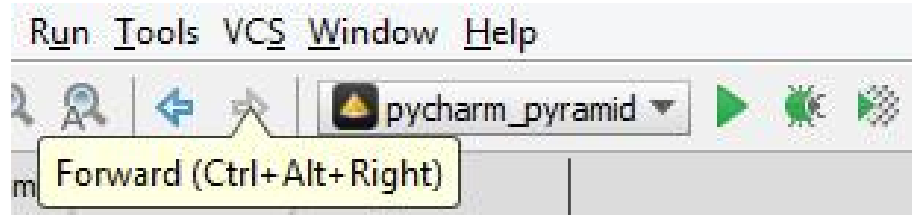
develop

install package in 'development mode'

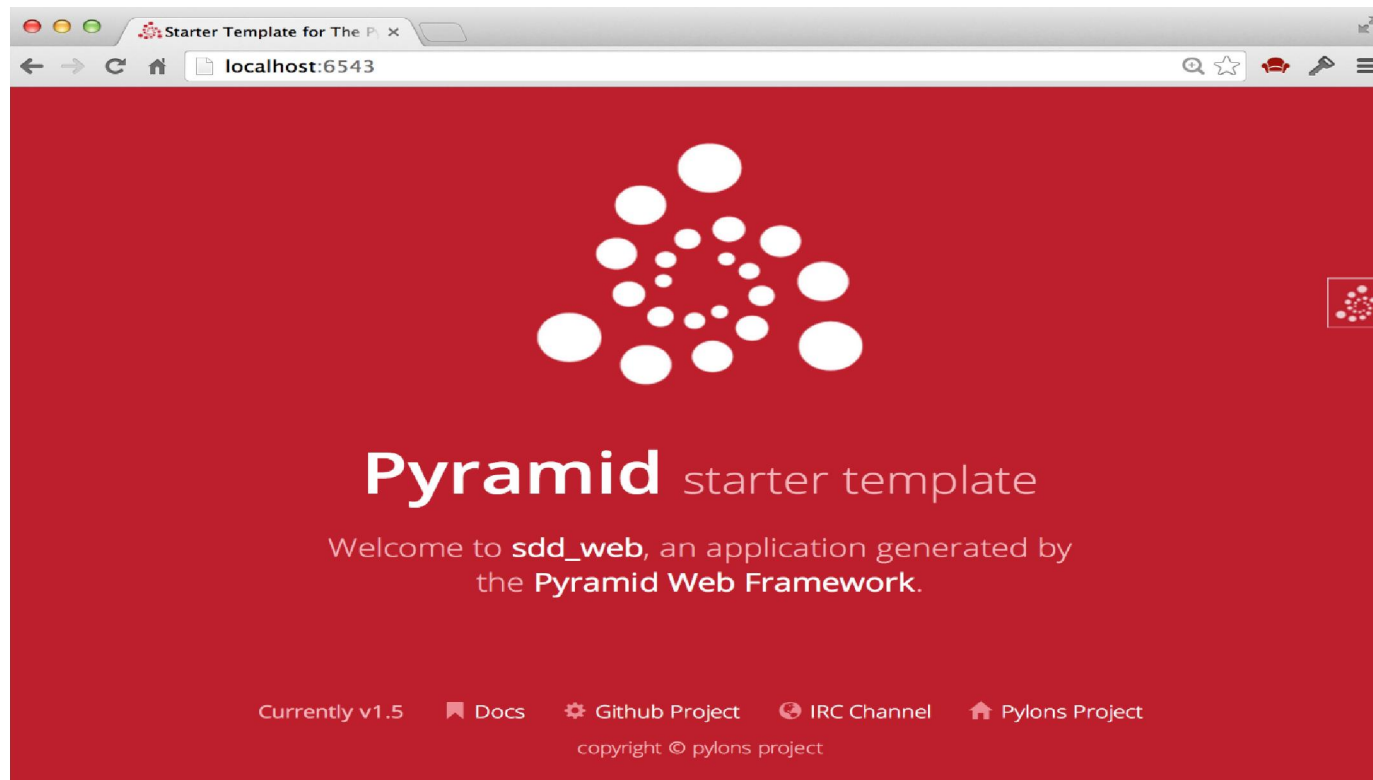


Run with no additional command arguments

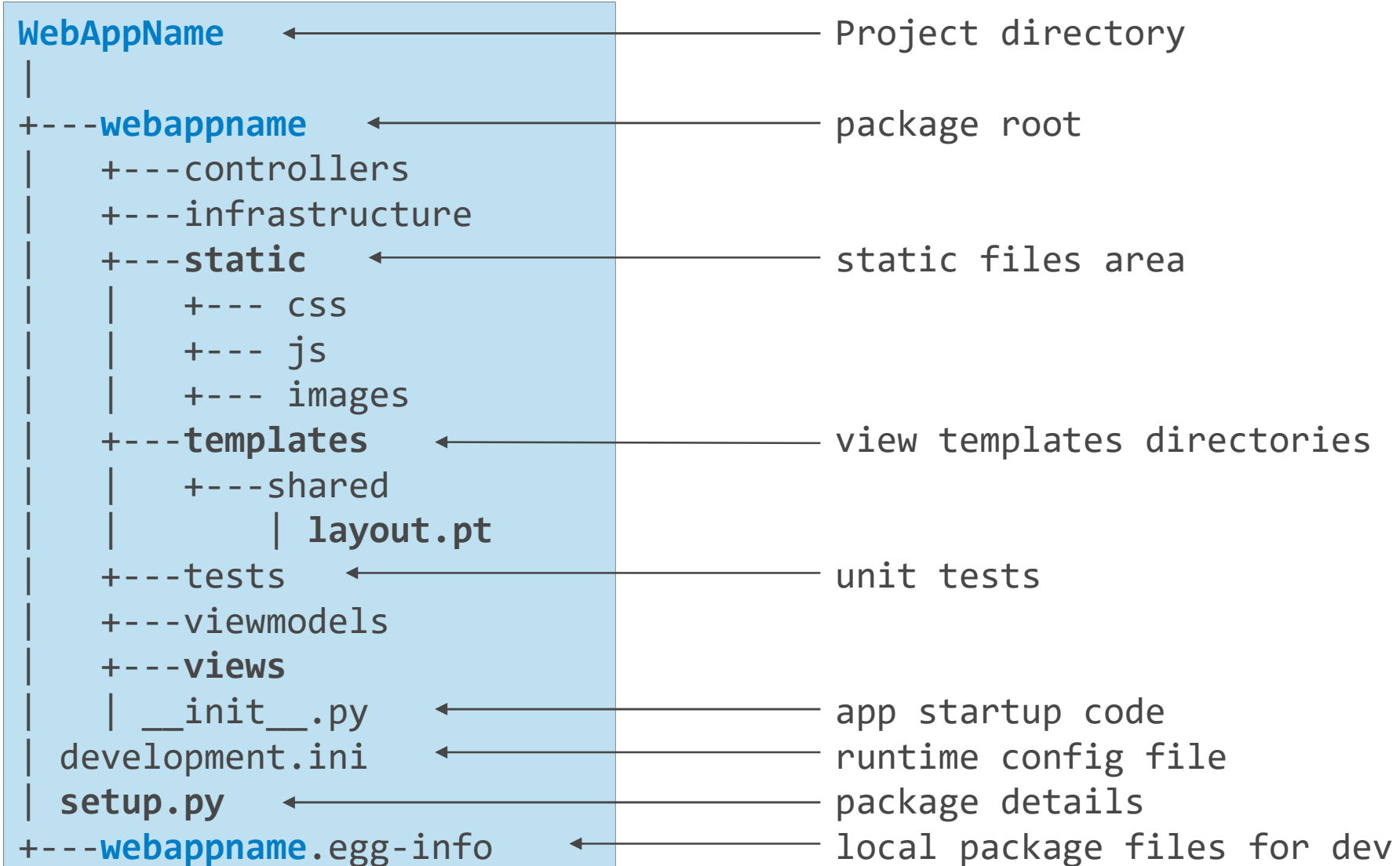
Getting started [Installing Pyramid via PyCharm]



5. Run the web app



File structure



Configure the server to run [config files]

- Configuration typically happens in two places in Pyramid
 - config_file.ini: development.ini or production.ini
 - __init__.py

```
# development.ini

[app:main]
use = egg:mywebapp

pyramid.reload_templates = true
pyramid.debug_authorization = false
pyramid.debug_notfound = false
pyramid.debug_routematch = false
pyramid.default_locale_name = en
pyramid.includes = pyramid_debugtoolbar

[server:main]
use = egg:waitress#main
host = 0.0.0.0
port = 6543
```

Configure the server to run [startup code]

- Configuration typically happens in two places in Pyramid
 - config_file.ini
 - __init__.py

```
# __init__.py
from pyramid.config import Configurator

def main(global_config, **settings):

    config = Configurator(settings=settings)
    config.add_static_view('static', 'static',
                          cache_max_age=3600)
    config.add_route('home', '/')
    config.scan()

    return config.make_wsgi_app()
```

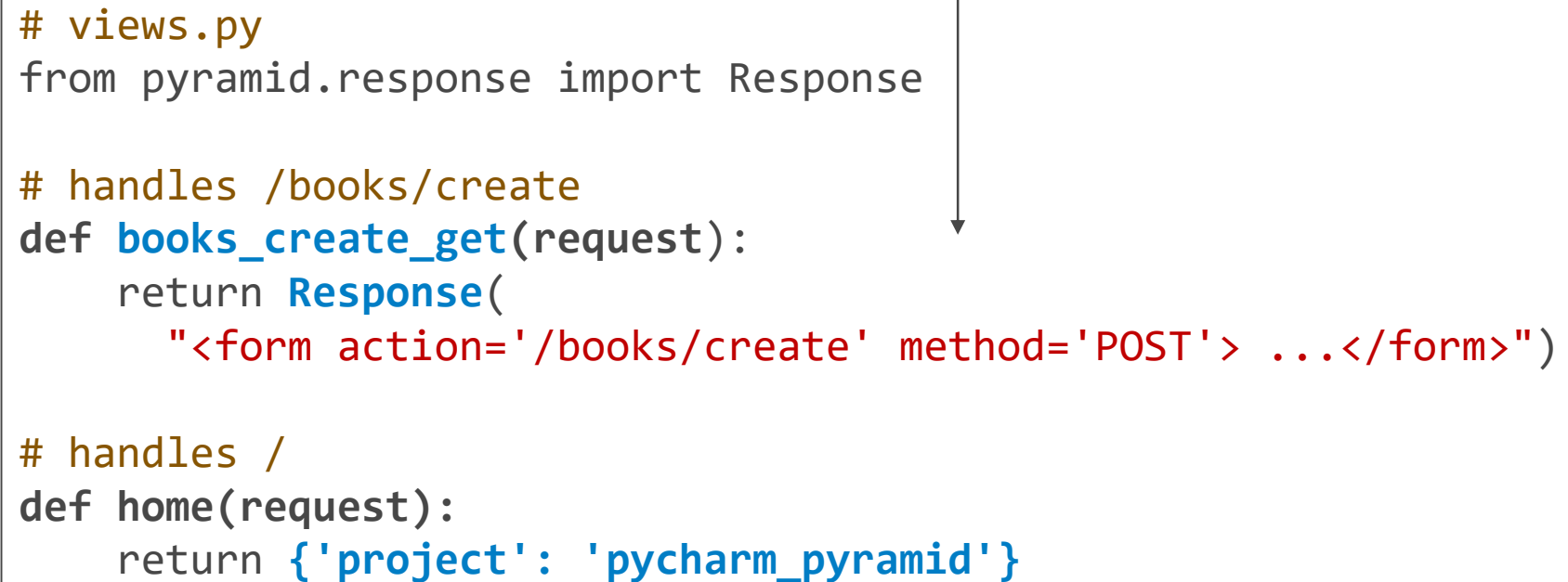
Views [individual methods]

- Views processing
 - **views (view methods)**

```
# views.py
from pyramid.response import Response

# handles /books/create
def books_create_get(request):
    return Response(
        "<form action='/books/create' method='POST'> ...</form>")

# handles /
def home(request):
    return {'project': 'pycharm_pyramid'}
```



can return raw HTML

Preferably return a 'model' dictionary + execute a template

Routes [defining routes]

- Routes map URLs (and data) to
 - views (view methods)
 - view classes (classes dedicated to a single view)

Each unique URL has a
dedicate route

```
# __init__.py

config = Configurator(settings=settings)
config.add_route('home', '/')
config.add_route('create_book', '/books/create')
config.add_route('edit_book', '/books/edit/{book_id}')
# ...
config.scan()
```

↑
Wire views to
routes in all py
files

↑
Route names
must be unique

↑
Data placeholders allow for data in the URL:

/books/edit/200
/books/edit/201
...

Routes [mapping to views]

- Routes map URLs (and data) to
 - views (**view methods**)
 - view classes (classes dedicated to a single view)

view_config decorator attaches view to routes

```
@view_config(route_name='home',  
             renderer='templates/mytemplate.pt')  
def home(request):  
    # ...  
  
@view_config(route_name='create_book',  
             request_method='POST')  
def books_create_post(request):  
    # ...
```

view_config selects template and can filter by HTTP verbs (among other values)

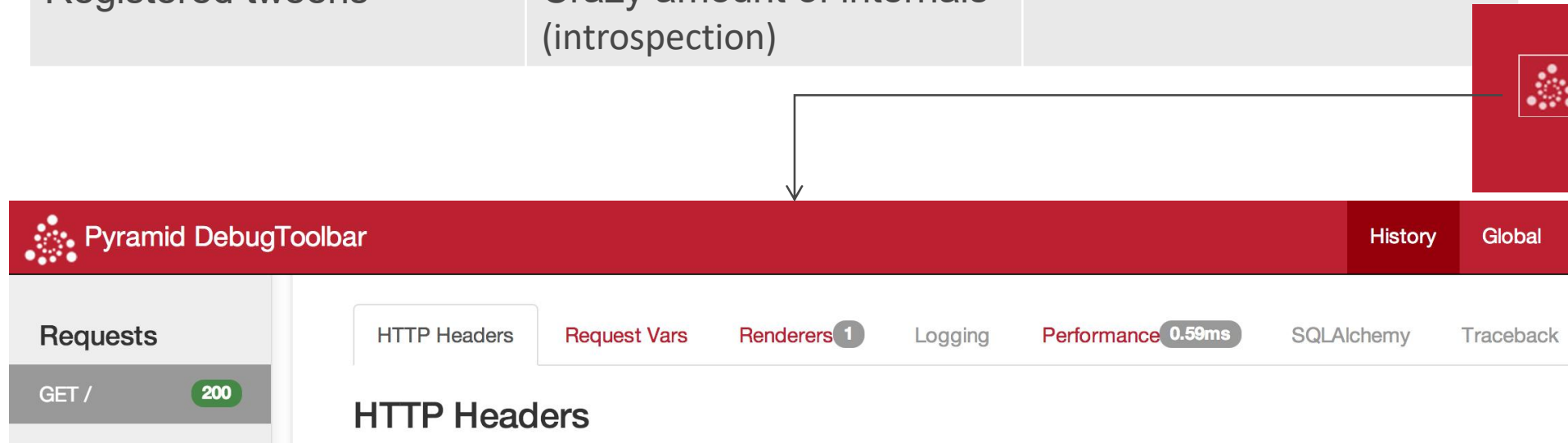
Chameleon Templates

- Pyramid supports multiple view templates
- Three
 - Jinja2
 - Chameleon
 - Mako

Debug Toolbar

- Pyramid comes with a fantastic debug toolbar
 - only visible when viewing the site locally
 - can be overridden

Details on all dependencies	Values from config.ini	Request/response details
get / post / cookies	Template details	App level log messages
Request times and profiler	Details about all routes	SQLAlchemy details
Registered tweens	Crazy amount of internals (introspection)	



Pyramid DebugToolbar

History Global

Requests

GET / 200

HTTP Headers Request Vars Renderers 1 Logging Performance 0.59ms SQLAlchemy Traceback

HTTP Headers

Summary

- Pyramid is a very flexible micro framework
- Pyramid can be installed as a package (with dependencies)
- Organizing your file structure will help as your site grows
- Jinja2 template allow rich HTML output