if !exists('g:test#python#frappe#file\_pattern')

let g:test#python#frappe#file\_pattern = '\v^test.\*\.py$'

endif

" Returns true if the given file belongs to your test runner

function! test#python#frappe#test\_file(file) abort

if fnamemodify(a:file, ':t') =~# g:test#python#frappe#file\_pattern

if exists('g:test#python#runner')

return g:test#python#runner ==# 'frappe'

else

return executable('bench')

endif

endif

endfunction

" Returns test runner's arguments which will run the current file and/or line

function! test#python#frappe#build\_position(type, position) abort

let path = s:get\_import\_path(a:position['file'])

let sitename = g:test#python#frappe#testsite

let module\_cmd = '--site ' . sitename . ' run-tests --module ' . path

if a:type ==# 'nearest'

let name = s:nearest\_test(a:position)

if !empty(name)

let test\_cmd = module\_cmd . ' --test ' . name

return [test\_cmd]

else

return [module\_cmd]

endif

elseif a:type ==# 'file'

return [module\_cmd]

else

return []

endif

endfunction

" Returns processed args (if you need to do any processing)

function! test#python#frappe#build\_args(args) abort

return a:args

endfunction

function! s:get\_import\_path(filepath) abort

" Get path to file from cwd and without extension. rooter required for correct root.

let path = fnamemodify(a:filepath, ':.:r')

let path = substitute(path, '\/', '.', 'g')

return path

endfunction

function! s:nearest\_test(position) abort

let name = test#base#nearest\_test(a:position, g:test#python#patterns)

return join(name['test'], '.')

endfunction

" Returns the executable of your test runner

function! test#python#frappe#executable() abort

return 'bench'

endfunction