

clj-refactor Cheat Sheet by bilus via cheatography.com/23535/cs/5250/

Namespaces	
ai	add import
am	add a missing libspec
ar	add require
au	add use
cn	clean namespace
mf	move forms to another namespace
rr	remove unused requires
ru	convert :use's to :require's
sn	sort ns form
sr	stop referring
rd	remove (debug) function invocations

Hydra menus	
hh	parent menu
hn	namespace refactorings
hc	code refactorings
hp	project refactorings
ht	top-level form refactorings

Project	
ар	add and hotload dependency
hd	hotload dependency
рс	run project clean functions
rf	remove file or dir
sp	sort dependencies
up	update dependencies

as add stubs cc cycle surrounding collection type ci cycle between if and if-not cp cycle privacy defn/defn- dk destructure keys ec extract constant ed extract def ef extract function el expand let fe create function from example stub fu find usages il introduce let is inline symbol mf move form(s) to another ns ml move to let pf promote function literal/fn/defn rl remove let rs rename symbol tf wrap in -> th thread another expression tl wrap in ->> ua fully unwind threaded expression uw unwind threaded expression	Symbols		
ci cycle between if and if-not cp cycle privacy defn/defn- dk destructure keys ec extract constant ed extract def ef extract function el expand let fe create function from example stub fu find usages il introduce let is inline symbol mf move form(s) to another ns ml move to let pf promote function literal/fn/defn rl remove let rs rename symbol tf wrap in -> th thread another expression tl wrap in ->> ua fully unwind threaded expression	as	add stubs	
cp cycle privacy defn/defn- dk destructure keys ec extract constant ed extract def ef extract function el expand let fe create function from example stub fu find usages il introduce let is inline symbol mf move form(s) to another ns ml move to let pf promote function literal/fn/defn rl remove let rs rename symbol tf wrap in -> th thread another expression tl wrap in ->> ua fully unwind threaded expression	CC	cycle surrounding collection type	
c extract constant ed extract def ef extract function el expand let fe create function from example stub fu find usages il introduce let is inline symbol mf move form(s) to another ns ml move to let pf promote function literal/fn/defn rl remove let rs rename symbol tf wrap in -> th thread another expression tl wrap in ->> ua fully unwind threaded expression	ci	cycle between if and if-not	
ec extract constant ed extract def ef extract function el expand let fe create function from example stub fu find usages il introduce let is inline symbol mf move form(s) to another ns ml move to let pf promote function literal/fn/defn rl remove let rs rename symbol tf wrap in -> th thread another expression tl wrap in ->> ua fully unwind threaded expression	ср	cycle privacy defn/defn-	
ed extract def ef extract function el expand let fe create function from example stub fu find usages il introduce let is inline symbol mf move form(s) to another ns ml move to let pf promote function literal/fn/defn rl remove let rs rename symbol tf wrap in -> th thread another expression tl wrap in ->> ua fully unwind threaded expression	dk	destructure keys	
ef extract function el expand let fe create function from example stub fu find usages il introduce let is inline symbol mf move form(s) to another ns ml move to let pf promote function literal/fn/defn rl remove let rs rename symbol tf wrap in -> th thread another expression tl wrap in ->> ua fully unwind threaded expression	ec	extract constant	
el expand let fe create function from example stub fu find usages il introduce let is inline symbol mf move form(s) to another ns ml move to let pf promote function literal/fn/defn rl remove let rs rename symbol tf wrap in -> th thread another expression tl wrap in ->> ua fully unwind threaded expression	ed	extract def	
fe create function from example stub fu find usages il introduce let is inline symbol mf move form(s) to another ns ml move to let pf promote function literal/fn/defn rl remove let rs rename symbol tf wrap in -> th thread another expression tl wrap in ->> ua fully unwind threaded expression	ef	extract function	
fu find usages il introduce let is inline symbol mf move form(s) to another ns ml move to let pf promote function literal/fn/defn rl remove let rs rename symbol tf wrap in -> th thread another expression tl wrap in ->> ua fully unwind threaded expression	el	expand let	
il introduce let is inline symbol mf move form(s) to another ns ml move to let pf promote function literal/fn/defn rl remove let rs rename symbol tf wrap in -> th thread another expression tl wrap in ->> ua fully unwind threaded expression	fe	create function from example stub	
is inline symbol mf move form(s) to another ns ml move to let pf promote function literal/fn/defn rl remove let rs rename symbol tf wrap in -> th thread another expression tl wrap in ->> ua fully unwind threaded expression	fu	find usages	
mf move form(s) to another ns ml move to let pf promote function literal/fn/defn rl remove let rs rename symbol tf wrap in -> th thread another expression tl wrap in ->> ua fully unwind threaded expression	il	introduce let	
ml move to let pf promote function literal/fn/defn rl remove let rs rename symbol tf wrap in -> th thread another expression tl wrap in ->> ua fully unwind threaded expression	is	inline symbol	
pf promote function literal/fn/defn rl remove let rs rename symbol tf wrap in -> th thread another expression tl wrap in ->> ua fully unwind threaded expression	mf	move form(s) to another ns	
rl remove let rs rename symbol tf wrap in -> th thread another expression tl wrap in ->> ua fully unwind threaded expression	ml	move to let	
rs rename symbol tf wrap in -> th thread another expression tl wrap in ->> ua fully unwind threaded expression	pf	promote function literal/fn/defn	
tf wrap in -> th thread another expression tl wrap in ->> ua fully unwind threaded expression	rl	remove let	
th thread another expression tl wrap in ->> ua fully unwind threaded expression	rs	rename symbol	
tl wrap in ->> ua fully unwind threaded expression	tf	wrap in ->	
ua fully unwind threaded expression	th	thread another expression	
	tl	wrap in ->>	
uw unwind threaded expression	ua	fully unwind threaded expression	
	uw	unwind threaded expression	

ар	add and hotload dependency
hd	hotload dependency
рс	run project clean functions
rf	remove file or dir
sp	sort dependencies
up	update dependencies
	<u> </u>



By bilus cheatography.com/bilus/ Published 23rd September, 2015. Last updated 24th September, 2015. Page 1 of 1.

Misc

describe refactoring

show clj-refactor changelog

Sponsored by CrosswordCheats.com Learn to solve cryptic crosswords! http://crosswordcheats.com