

Spring Roo - Reference Documentation

DISID Corporation S.L.

Table of Contents

Preface	2
1. Overview	2
2. Requirements	2
3. Download Spring Roo	2
4. Install Spring Roo	2
5. Command Index	3
Addon Suite Commands	4
addon suite install name	4
addon suite install url	4
addon suite list	4
addon suite start	4
addon suite stop	5
addon suite uninstall	5
Audit Commands	6
audit add	6
audit setup	6
Backup Commands	7
backup	7
Classpath Commands	8
class	8
constructor	8
enum constant	9
enum type	9
focus	10
interface	10
Controller Commands	11
web mvc controller	11
web mvc setup	12
Creator Commands	13
addon create advanced	13
addon create i18n	13
addon create simple	14
addon create suite	14
addon create wrapper	15
Data On Demand Commands	16
dod	16
dod	17
Dto Commands	18
dto	18
Equals Commands	19
equals	19
Felix Delegator	20

lg	20
exit	20
Field Commands	21
field boolean	21
field date	22
field embedded	23
field enum	24
field file	24
field list	25
field number	27
field other	28
field reference	29
field set	30
field string	32
Finder Commands	34
finder add	34
Help Commands	35
help	35
reference guide	35
Hint Commands	36
hint	36
I18n Commands	37
web mvc language	37
Integration Test Commands	38
test integration	38
J Line Shell Component	39
date	39
script	39
system properties	40
version	40
Jpa Commands	41
embeddable	41
entity jpa	41
jpa setup	43
Maven Commands	45
dependency add	45
dependency remove	45
maven repository add	46
maven repository remove	46
module create	46
module focus	47
perform assembly	47
perform clean	47
perform command	47

perform eclipse	48
perform package	48
perform tests	48
Metadata Commands	49
metadata cache	49
metadata for id	49
metadata for module	49
metadata for type	49
metadata status	50
metadata trace	50
Obr Add On Commands	51
addon info bundle	51
addon install bundle	51
addon install url	51
addon list	51
addon remove	52
addon search	52
Obr Repository Commands	53
addon repository add	53
addon repository introspect	53
addon repository list	53
addon repository remove	53
Os Commands	54
Pgp Commands	55
pgp automatic trust	55
pgp key view	55
pgp list trusted keys	55
pgp refresh all	55
pgp status	56
pgp trust	56
pgp untrust	56
Process Manager Diagnostics Listener	57
process manager debug	57
Project Commands	58
development mode	58
project scan now	58
project scan speed	58
project scan status	58
project setup	59
Prop File Commands	60
property add	60
property list	60
property remove	60
Push In Commands	61

push-in	61
Repository Jpa Commands	62
repository jpa	62
Security Commands	63
security setup	63
Service Commands	64
service	64
Settings Commands	65
settings add	65
settings list	65
settings remove	65
Unit Test Commands	66
test unit	66
View Commands	67
web mvc templates setup	67
web mvc view setup	67
Web Finder Commands	68
web mvc finder	68
6. Roo Resources	68
Spring Roo Project Home Page	68
Downloads and Maven Repositories	69
StackOverFlow	69
Twitter	70
Issue Tracking	70
Source Repository	71
Commercial Products and Services	71
Other	71

2.0.0.M2

© 2016 ***The original authors.***

Copies of this document may be made for your own use and for distribution to others, provided that you do not charge any fee for such copies and further provided that each copy contains this Copyright Notice, whether distributed in print or electronically.

Preface

1. Overview

Spring Roo is an easy-to-use development tool for quickly building web applications in the Java programming language, which can be used as an standalone application or as an Eclipse or STS plugin. It allows you to build high-quality, high-performance, lock-in-free enterprise applications in just minutes.

Best of all, Roo works alongside your existing Java and Spring knowledge, skills and experience. You probably won't need to learn anything new to use Roo, as there's no new language or runtime platform needed. You simply program in your normal Java way and Roo just works, sitting in the background taking care of the things you don't want to worry about.

2. Requirements

To get started, please ensure you have the following system dependencies:

- A Linux, Apple or Windows-based operating system (other operating systems may work but are not guaranteed).
- A [Java JDK 6](#) or newer installed. Java JDK 7 is recommended.
- [Apache Maven 3.0](#) or above installed and in the path.

We always recommend you use the latest version of Java and Maven that are available for your platform. We also recommend that you use [Spring Tool Suite \(STS\)](#) which includes a number of features that make working with Roo even easier (you can of course use Roo without an IDE at all if you prefer).

3. Download Spring Roo

You can download the current release from Spring Roo project page [downloads section](#).

You can also build a distribution ZIP yourself from our [source control repository](#).

4. Install Spring Roo

Once you have satisfied the initial requirements, you can install Roo by following these steps:

1. Unzip the distribution which will unpack to a single installation directory; this will be known as \$ROO_HOME in the directions below
2. If using Windows, add \$ROO_HOME\bin to your **PATH** environment variable

3. If using Linux or Apple, create a symbolic link using a command such as `sudo ln -s $ROO_HOME/bin/roo.sh /usr/bin/roo`

Next verify Roo has been installed correctly. This can be done using the following commands:

```
$ mkdir roo-test
$ cd roo-test
$ roo quit

      _ _ _ _ _
     /  _  \ /  _  \ /  _  \
    /  _/ / / / / / / / /
   /  _/ / / / / / / / /
  /  _/ / / / / / / / /
 /  _/ | | \ _ _ _ / \ _ _ _ /   W.X.Y.ZZ [rev RRR]

Welcome to Spring Roo. For assistance press TAB or type "hint" then hit ENTER.
$ cd ..
$ rmdir roo-test
```

If you see the logo appear, you've installed Roo successfully. For those curious, the "[rev RRR]" refers to the Git commit ID used to compile that particular build of Roo.

5. Command Index

This appendix was automatically built from Roo 2.0.0.M2

Commands are listed in alphabetic order, and are shown in monospaced font with any mandatory options you must specify when using the command. Most commands accept a large number of options, and all of the possible options for each command are presented in this appendix.

Addon Suite Commands

Addon Suite Commands are contained in `org.springframework.roo.addon.suite.AddonSuiteCommands`.

addon suite install name

Install some 'Roo Addon Suite' from installed OBR Repository

```
addon suite install name --symbolicName
```

--symbolicName

Name that identifies the 'Roo Addon Suite'; default: '*NULL*' (mandatory)

addon suite install url

Install some 'Roo Addon Suite' from URL

```
addon suite install url --url
```

--url

URL of Roo Addon Suite .esa file; default: '*NULL*' (mandatory)

addon suite list

Lists all installed 'Roo Addon Suite'. If you want to list all available 'Roo Addon Suites' on Repository, use `--repository` parameter

```
addon suite list
```

--repository

OBR Repository where the 'Roo Addon Suite' are located; default: '*NULL*'

addon suite start

Start some installed 'Roo Addon Suite'

```
addon suite start --symbolicName
```

--symbolicName

Name that identifies the 'Roo Addon Suite'; default: '*NULL*' (mandatory)

addon suite stop

Stop some started 'Roo Addon Suite'

```
addon suite stop --symbolicName
```

--symbolicName

Name that identifies the 'Roo Addon Suite'; default: '*NULL*' (mandatory)

addon suite uninstall

Uninstall some installed 'Roo Addon Suite'

```
addon suite uninstall --symbolicName
```

--symbolicName

Name that identifies the 'Roo Addon Suite'; default: '*NULL*' (mandatory)

Audit Commands

Audit Commands are contained in `org.springframework.roo.addon.security.addon.audit.AuditCommands`.

audit add

Adds support for auditing a JPA entity

```
audit add --entity --createdDateColumn --modifiedDateColumn --createdByColumn  
--modifiedByColumn
```

--entity

The entity which should be audited; default: *'NULL'* (mandatory)

--createdDateColumn

The DB column used for storing created date info; default: *'NULL'* (mandatory)

--modifiedDateColumn

The DB column used for storing modified date info; default: *'NULL'* (mandatory)

--createdByColumn

The DB column used for storing created by info; default: *'NULL'* (mandatory)

--modifiedByColumn

The DB column used for storing modified by info; default: *'NULL'* (mandatory)

audit setup

Install audit support into your project

```
audit setup --package
```

--package

The package in which new classes needed for audit will be placed. Note that module will be ignored as new classes will be installed in app's config module/s; default: *'NULL'* (mandatory)

Backup Commands

Backup Commands are contained in `org.springframework.roo.addon.backup.BackupCommands`.

backup

Backup your project to a zip file

```
backup
```

This command does not accept any options.

Classpath Commands

Classpath Commands are contained in
org.springframework.roo.classpath.operations.ClasspathCommands.

class

Creates a new Java class source file in any project path

```
class --class
```

--class

The name of the class to create; default: *'NULL'* (mandatory)

--rooAnnotations

Whether the generated class should have common Roo annotations; default if option present: 'true';
default if option not present: 'false'

--path

Source directory to create the class in; default: 'FOCUSED:SRC_MAIN_JAVA'

--extends

The superclass (defaults to java.lang.Object); default if option not present: 'java.lang.Object'

--implements

The interface to implement; default: *'NULL'*

--abstract

Whether the generated class should be marked as abstract; default if option present: 'true'; default if
option not present: 'false'

--permitReservedWords

Indicates whether reserved words are ignored by Roo; default if option present: 'true'; default if
option not present: 'false'

constructor

Creates a class constructor

```
constructor
```

--class

The name of the class to receive this constructor; default if option not present: '*'

--fields

The fields to include in the constructor. Multiple field names must be a double-quoted list separated by spaces

enum constant

Inserts a new enum constant into an enum

```
enum constant --name
```

--class

The name of the enum class to receive this field; default if option not present: '*'

--name

The name of the constant; default: 'NULL' (mandatory)

--permitReservedWords

Indicates whether reserved words are ignored by Roo; default if option present: 'true'; default if option not present: 'false'

enum type

Creates a new Java enum source file in any project path

```
enum type --class
```

--class

The name of the enum to create; default: 'NULL' (mandatory)

--path

Source directory to create the enum in; default: 'FOCUSED:SRC_MAIN_JAVA'

--permitReservedWords

Indicates whether reserved words are ignored by Roo; default if option present: 'true'; default if option not present: 'false'

focus

Changes focus to a different type

```
focus --class
```

--class

The type to focus on; default: '*NULL*' (mandatory)

interface

Creates a new Java interface source file in any project path

```
interface --class
```

--class

The name of the interface to create; default: '*NULL*' (mandatory)

--path

Source directory to create the interface in; default: 'FOCUSED:SRC_MAIN_JAVA'

--permitReservedWords

Indicates whether reserved words are ignored by Roo; default if option present: 'true'; default if option not present: 'false'

Controller Commands

Controller Commands are contained in
org.springframework.roo.addon.web.mvc.controller.addon.ControllerCommands.

web mvc controller

Generates new @RooController inside current project

```
web mvc controller --package --entity --service --path
```

--all

Indicates if developer wants to generate controllers for every entity of current project ; default if option present: 'true'; default if option not present: 'false'

--package

This param will be mandatory if --all parameter has been specified. Indicates which package should be used to include generated controllers; default: 'NULL' (mandatory)

--controller

Indicates the new controller class to generate. Also, you can indicates an existing controller to update it.; default: 'NULL'

--entity

This param will be mandatory if --controller parameter has been specified with a new controller to generate. Indicates the entity that new controller will be manage.; default: 'NULL' (mandatory)

--service

This param will be mandatory if --entity parameter has been specified. Indicates the service that new controller will use to access to negotiation layer.; default: 'NULL' (mandatory)

--path

Indicates @RequestMapping to be used on this controller. Is not necessary to specify '/'. Spring Roo shell will include it automatically.; default: 'NULL' (mandatory)

--responseType

Indicates the responseType to be used by generated controller. Depending of the selected responseType, generated methods and views will vary.; default: 'JSON'

--formattersPackage

Indicates project package where formatters should be generated. By default they will be generated inside the same controllers package.; default: 'NULL'

web mvc setup

Includes Spring MVC on generated project

```
web mvc setup --module
```

--module

The application module where to install the persistence; default if option not present: '' (mandatory)

--appServer

The server where deploy the application; default if option not present: 'EMBEDDED'

Creator Commands

Creator Commands are contained in `org.springframework.roo.addon.creator.CreatorCommands`.

addon create advanced

Create a new advanced add-on for Spring Roo (commands + operations metadata + trigger annotation + dependencies)

```
addon create advanced --topLevelPackage
```

--topLevelPackage

The top level package of the new addon; default: *'NULL'* (mandatory)

--description

Description of your addon (surround text with double quotes); default: *'NULL'*

--projectName

Provide a custom project name (if not provided the top level package name will be used instead); default: *'NULL'*

addon create i18n

Create a new Internationalization add-on for Spring Roo

```
addon create i18n --topLevelPackage --locale --messageBundle
```

--topLevelPackage

The top level package of the new addon; default: *'NULL'* (mandatory)

--locale

The locale abbreviation (ie: en, or more specific like en_AU, or de_DE); default: *'NULL'* (mandatory)

--messageBundle

Fully qualified path to the messages_xx.properties file; default: *'NULL'* (mandatory)

--language

The full name of the language (used as a label for the UI); default: *'NULL'*

--flagGraphic

Fully qualified path to flag xx.png file; default: *'NULL'*

--description

Description of your addon (surround text with double quotes); default: *'NULL'*

--projectName

Provide a custom project name (if not provided the top level package name will be used instead); default: *'NULL'*

addon create simple

Create a new simple add-on for Spring Roo (commands + operations)

```
addon create simple --topLevelPackage
```

--topLevelPackage

The top level package of the new addon; default: *'NULL'* (mandatory)

--description

Description of your addon (surround text with double quotes); default: *'NULL'*

--projectName

Provide a custom project name (if not provided the top level package name will be used instead); default: *'NULL'*

addon create suite

Create a new Spring Roo Addon Suite for Spring Roo (two sample addons repository + suite generator)

```
addon create suite --topLevelPackage
```

--topLevelPackage

The top level package of all Spring Roo Addon Suite; default: *'NULL'* (mandatory)

--description

Description of your Roo Addon Suite (surround text with double quotes); default: *'NULL'*

--projectName

Provide a custom project name (if not provided the top level package name will be used instead); default: *'NULL'*

addon create wrapper

Create a new add-on for Spring Roo which wraps a maven artifact to create a OSGi compliant bundle

```
addon create wrapper --topLevelPackage --groupId --artifactId --version --vendorName  
--licenseUrl
```

--topLevelPackage

The top level package of the new wrapper bundle; default: *'NULL'* (mandatory)

--groupId

Dependency group id; default: *'NULL'* (mandatory)

--artifactId

Dependency artifact id; default: *'NULL'* (mandatory)

--version

Dependency version; default: *'NULL'* (mandatory)

--vendorName

Dependency vendor name; default: *'NULL'* (mandatory)

--licenseUrl

Dependency license URL; default: *'NULL'* (mandatory)

--docUrl

Dependency documentation URL; default: *'NULL'*

--description

Description of the bundle (use keywords with #-tags for better search integration); default: *'NULL'*

--projectName

Provide a custom project name (if not provided the top level package name will be used instead); default: *'NULL'*

--osgiImports

Contents of Import-Package in OSGi manifest; default: *'NULL'*

Data On Demand Commands

Data On Demand Commands are contained in
`org.springframework.roo.addon.dod.addon.DataOnDemandCommands`.

dod

Creates a new data on demand for the specified entity

dod

--entity

The entity which this data on demand class will create and modify as required; default if option not present: '*'

--class

The class which will be created to hold this data on demand provider (defaults to the entity name + 'DataOnDemand'); default: *NULL*

--permitReservedWords

Indicates whether reserved words are ignored by Roo; default if option present: 'true'; default if option not present: 'false'

Dto Commands

Dto Commands are contained in `org.springframework.roo.addon.dto.addon.DtoCommands`.

dto

Creates a new DTO class in SRC_MAIN_JAVA

```
dto --class
```

--class

Name of the DTO class to create, including package and module (if multimodule project); default: *'NULL'* (mandatory)

--entity

Name of the entity which can be used to create DTO from; default: *'NULL'*

--all

Whether should create one DTO per each entity in the project; default if option present: *'true'*; default if option not present: *'false'*

--fields

Comma separated list of entity fields to be included into the DTO; default: *'NULL'*

--excludeFields

Comma separated list of entity fields to be excluded into the DTO; default: *'NULL'*

--immutable

Whether the DTO should be immutable; default if option present: *'true'*; default if option not present: *'false'*

--utilityMethods

Whether the DTO should implement *'toString'*, *'hashCode'* and *'equals'* methods; default if option present: *'true'*; default if option not present: *'false'*

--serializable

Whether the DTO should implement *Serializable*; default if option present: *'true'*; default if option not present: *'false'*

Equals Commands

Equals Commands are contained in
org.springframework.roo.addon.javabean.addon.EqualsCommands.

equals

Add equals and hashCode methods to a class

```
equals
```

--class

The name of the class; default if option not present: '*'

--appendSuper

Whether to call the super class equals and hashCode methods; default if option present: 'true';
default if option not present: 'false'

--excludeFields

The fields to exclude in the equals and hashCode methods. Multiple field names must be a double-quoted list separated by spaces

Felix Delegator

Felix Delegator are contained in `org.springframework.roo.felix.FelixDelegator`.

!g

Passes a command directly through to the Felix shell infrastructure

!g

--[default]

The command to pass to Felix (WARNING: no validation or security checks are performed); default: 'help'

exit

Exits the shell

```
exit
```

This command does not accept any options.

Field Commands

Field Commands are contained in `org.springframework.roo.addon.field.addon.FieldCommands`.

field boolean

Adds a private boolean field to an existing Java source file

```
field boolean --fieldName --class --column
```

--fieldName

The name of the field to add; default: *'NULL'* (mandatory)

--class

The name of the class to receive this field; default if option not present: *'*'* (mandatory)

--notNull

Whether this value cannot be null; default if option present: *'true'*; default if option not present: *'false'*

--nullRequired

Whether this value must be null; default if option present: *'true'*; default if option not present: *'false'*

--assertFalse

Whether this value must assert false; default if option present: *'true'*; default if option not present: *'false'*

--assertTrue

Whether this value must assert true; default if option present: *'true'*; default if option not present: *'false'*

--column

The JPA `@Column` name; default: *'NULL'* (mandatory)

--value

Inserts an optional Spring `@Value` annotation with the given content; default: *'NULL'*

--comment

An optional comment for JavaDocs; default: *'NULL'*

--primitive

Indicates to use a primitive type; default if option present: *'true'*; default if option not present: *'false'*

--transient

Indicates to mark the field as transient; default if option present: 'true'; default if option not present: 'false'

--permitReservedWords

Indicates whether reserved words are ignored by Roo; default if option present: 'true'; default if option not present: 'false'

field date

Adds a private date field to an existing Java source file

```
field date --fieldName --type --class --column
```

--fieldName

The name of the field to add; default: 'NULL' (mandatory)

--type

The Java type of the entity; default: 'NULL' (mandatory)

--persistenceType

The type of persistent storage to be used; default: 'NULL'

--class

The name of the class to receive this field; default if option not present: '*' (mandatory)

--notNull

Whether this value cannot be null; default if option present: 'true'; default if option not present: 'false'

--nullRequired

Whether this value must be null; default if option present: 'true'; default if option not present: 'false'

--future

Whether this value must be in the future; default if option present: 'true'; default if option not present: 'false'

--past

Whether this value must be in the past; default if option present: 'true'; default if option not present: 'false'

--column

The JPA @Column name; default: 'NULL' (mandatory)

--comment

An optional comment for JavaDocs; default: *'NULL'*

--value

Inserts an optional Spring @Value annotation with the given content; default: *'NULL'*

--transient

Indicates to mark the field as transient; default if option present: *'true'*; default if option not present: *'false'*

--permitReservedWords

Indicates whether reserved words are ignored by Roo; default if option present: *'true'*; default if option not present: *'false'*

--dateFormat

Indicates the style of the date format (ignored if dateTimeFormatPattern is specified); default: *'MEDIUM'*

--timeFormat

Indicates the style of the time format (ignored if dateTimeFormatPattern is specified); default: *'NONE'*

--dateTimeFormatPattern

Indicates a DateTime format pattern such as yyyy-MM-dd hh:mm:ss a; default: *'NULL'*

field embedded

Adds a private @Embedded field to an existing Java source file

```
field embedded --fieldName --type --class
```

--fieldName

The name of the field to add; default: *'NULL'* (mandatory)

--type

The Java type of the @Embeddable class; default: *'NULL'* (mandatory)

--class

The name of the @Entity class to receive this field; default if option not present: *'*'* (mandatory)

--permitReservedWords

Indicates whether reserved words are ignored by Roo; default if option present: *'true'*; default if option not present: *'false'*

field enum

Adds a private enum field to an existing Java source file

```
field enum --fieldName --type --class --column
```

--fieldName

The name of the field to add; default: *'NULL'* (mandatory)

--type

The enum type of this field; default: *'NULL'* (mandatory)

--class

The name of the class to receive this field; default if option not present: *'*'* (mandatory)

--column

The JPA @Column name; default: *'NULL'* (mandatory)

--notNull

Whether this value cannot be null; default if option present: *'true'*; default if option not present: *'false'*

--nullRequired

Whether this value must be null; default if option present: *'true'*; default if option not present: *'false'*

--enumType

The fetch semantics at a JPA level; default: *'NULL'*

--comment

An optional comment for JavaDocs; default: *'NULL'*

--transient

Indicates to mark the field as transient; default if option present: *'true'*; default if option not present: *'false'*

--permitReservedWords

Indicates whether reserved words are ignored by Roo; default if option present: *'true'*; default if option not present: *'false'*

field file

Adds a byte array field for storing uploaded file contents (JSF-scaffolded UIs only)

```
field file --fieldName --class --contentType --column
```

--fieldName

The name of the file upload field to add; default: *'NULL'* (mandatory)

--class

The name of the class to receive this field; default if option not present: *'*'* (mandatory)

--contentType

The content type of the file; default: *'NULL'* (mandatory)

--autoUpload

Whether the file is uploaded automatically when selected; default if option present: *'true'*; default if option not present: *'false'*

--column

The JPA @Column name; default: *'NULL'* (mandatory)

--notNull

Whether this value cannot be null; default if option present: *'true'*; default if option not present: *'false'*

--permitReservedWords

Indicates whether reserved words are ignored by Roo; default if option present: *'true'*; default if option not present: *'false'*

field list

Adds a private List field to an existing Java source file (eg the 'one' side of a many-to-one)

```
field list --fieldName --type --class --joinTable --joinColumns --referencedColumns  
--inverseJoinColumns --inverseReferencedColumns
```

--fieldName

The name of the field to add; default: *'NULL'* (mandatory)

--type

The entity which will be contained within the Set; default: *'NULL'* (mandatory)

--class

The name of the class to receive this field; default if option not present: *'*'* (mandatory)

--mappedBy

The field name on the referenced type which owns the relationship; default: *'NULL'*

--notNull

Whether this value cannot be null; default if option present: *'true'*; default if option not present: *'false'*

--nullRequired

Whether this value must be null; default if option present: *'true'*; default if option not present: *'false'*

--sizeMin

The minimum number of elements in the collection; default: *'NULL'*

--sizeMax

The maximum number of elements in the collection; default: *'NULL'*

--cardinality

The relationship cardinality at a JPA level; default: *'ONE_TO_MANY'*

--fetch

The fetch semantics at a JPA level; default: *'NULL'*

--comment

An optional comment for JavaDocs; default: *'NULL'*

--transient

Indicates to mark the field as transient; default if option present: *'true'*; default if option not present: *'false'*

--permitReservedWords

Indicates whether reserved words are ignored by Roo; default if option present: *'true'*; default if option not present: *'false'*

--joinTable

Join table name. Most usually used in @ManyToMany relations. If name not specified it will take default @JoinTable name value.; default if option present: *'ROO_JOIN_TABLE'* (mandatory)

--joinColumns

Comma separated list of join table's foreign key columns which references the table of the entity owning the relation; default: *'NULL'* (mandatory)

--referencedColumns

Comma separated list of foreign key referenced columns in the table of the entity owning the relation; default: *'NULL'* (mandatory)

--inverseJoinColumns

Comma separated list of join table's foreign key columns which references the table of the entity that does not own the relation; default: *'NULL'* (mandatory)

--inverseReferencedColumns

Comma separated list of foreign key referenced columns in the table of the entity that does not own the relation; default: *'NULL'* (mandatory)

--cascadeType

CascadeType. Possible values are ALL, DETACH, MERGE, PERSIST, REFRESH and REMOVE.; default: *'ALL'*

field number

Adds a private numeric field to an existing Java source file

```
field number --fieldName --type --class --column
```

--fieldName

The name of the field to add; default: *'NULL'* (mandatory)

--type

The Java type of the entity; default: *'NULL'* (mandatory)

--class

The name of the class to receive this field; default if option not present: *'*'* (mandatory)

--notNull

Whether this value cannot be null; default if option present: *'true'*; default if option not present: *'false'*

--nullRequired

Whether this value must be null; default if option present: *'true'*; default if option not present: *'false'*

--decimalMin

The BigDecimal string-based representation of the minimum value; default: *'NULL'*

--decimalMax

The BigDecimal string based representation of the maximum value; default: *'NULL'*

--digitsInteger

Maximum number of integral digits accepted for this number; default: *'NULL'*

--digitsFraction

Maximum number of fractional digits accepted for this number; default: 'NULL'

--min

The minimum value; default: 'NULL'

--max

The maximum value; default: 'NULL'

--column

The JPA @Column name; default: 'NULL' (mandatory)

--comment

An optional comment for JavaDocs; default: 'NULL'

--value

Inserts an optional Spring @Value annotation with the given content; default: 'NULL'

--transient

Indicates to mark the field as transient; default if option present: 'true'; default if option not present: 'false'

--primitive

Indicates to use a primitive type if possible; default if option present: 'true'; default if option not present: 'false'

--unique

Indicates whether to mark the field with a unique constraint; default if option present: 'true'; default if option not present: 'false'

--permitReservedWords

Indicates whether reserved words are ignored by Roo; default if option present: 'true'; default if option not present: 'false'

field other

Inserts a private field into the specified file

```
field other --fieldName --type --class --column
```

--fieldName

The name of the field; default: 'NULL' (mandatory)

--type

The Java type of this field; default: '*NULL*' (mandatory)

--class

The name of the class to receive this field; default if option not present: '*' (mandatory)

--notNull

Whether this value cannot be null; default if option present: 'true'; default if option not present: 'false'

--nullRequired

Whether this value must be null; default if option present: 'true'; default if option not present: 'false'

--comment

An optional comment for JavaDocs; default: '*NULL*'

--column

The JPA @Column name; default: '*NULL*' (mandatory)

--value

Inserts an optional Spring @Value annotation with the given content; default: '*NULL*'

--transient

Indicates to mark the field as transient; default if option present: 'true'; default if option not present: 'false'

--permitReservedWords

Indicates whether reserved words are ignored by Roo; default if option present: 'true'; default if option not present: 'false'

field reference

Adds a private reference field to an existing Java source file (eg the 'many' side of a many-to-one)

```
field reference --fieldName --type --class --joinColumnName
```

--fieldName

The name of the field to add; default: '*NULL*' (mandatory)

--type

The Java type of the entity to reference; default: '*NULL*' (mandatory)

--class

The name of the class to receive this field; default if option not present: '*' (mandatory)

--notNull

Whether this value cannot be null; default if option present: 'true'; default if option not present: 'false'

--nullRequired

Whether this value must be null; default if option present: 'true'; default if option not present: 'false'

--joinColumnName

The JPA @JoinColumn name; default: 'NULL' (mandatory)

--referencedColumnName

The JPA @JoinColumn referencedColumnName; default: 'NULL'

--cardinality

The relationship cardinality at a JPA level; default: 'MANY_TO_ONE'

--fetch

The fetch semantics at a JPA level; default: 'NULL'

--comment

An optional comment for JavaDocs; default: 'NULL'

--transient

Indicates to mark the field as transient; default if option present: 'true'; default if option not present: 'false'

--permitReservedWords

Indicates whether reserved words are ignored by Roo; default if option present: 'true'; default if option not present: 'false'

--cascadeType

CascadeType. Possible values are ALL, DETACH, MERGE, PERSIST, REFRESH and REMOVE.; default if option present: 'ALL'

field set

Adds a private Set field to an existing Java source file (eg the 'one' side of a many-to-one)

```
field set --fieldName --type --class --joinTable --joinColumns --referencedColumns
--inverseJoinColumns --inverseReferencedColumns
```

--fieldName

The name of the field to add; default: *'NULL'* (mandatory)

--type

The entity which will be contained within the Set; default: *'NULL'* (mandatory)

--class

The name of the class to receive this field; default if option not present: *'*'* (mandatory)

--mappedBy

The field name on the referenced type which owns the relationship; default: *'NULL'*

--notNull

Whether this value cannot be null; default if option present: *'true'*; default if option not present: *'false'*

--nullRequired

Whether this value must be null; default if option present: *'true'*; default if option not present: *'false'*

--sizeMin

The minimum number of elements in the collection; default: *'NULL'*

--sizeMax

The maximum number of elements in the collection; default: *'NULL'*

--cardinality

The relationship cardinality at a JPA level; default: *'ONE_TO_MANY'*

--fetch

The fetch semantics at a JPA level; default: *'NULL'*

--comment

An optional comment for JavaDocs; default: *'NULL'*

--transient

Indicates to mark the field as transient; default if option present: *'true'*; default if option not present: *'false'*

--permitReservedWords

Indicates whether reserved words are ignored by Roo; default if option present: *'true'*; default if option not present: *'false'*

--joinTable

Join table name. Most usually used in @ManyToMany relations. If name not specified it will take

default @JoinTable name value.; default if option present: *'ROO_JOIN_TABLE'* (mandatory)

--joinColumns

Comma separated list of join table's foreign key columns which references the table of the entity owning the relation; default: *'NULL'* (mandatory)

--referencedColumns

Comma separated list of foreign key referenced columns in the table of the entity owning the relation; default: *'NULL'* (mandatory)

--inverseJoinColumns

Comma separated list of join table's foreign key columns which references the table of the entity that does not own the relation; default: *'NULL'* (mandatory)

--inverseReferencedColumns

Comma separated list of foreign key referenced columns in the table of the entity that does not own the relation; default: *'NULL'* (mandatory)

--cascadeType

CascadeType. Possible values are ALL, DETACH, MERGE, PERSIST, REFRESH and REMOVE.; default: *'ALL'*

field string

Adds a private string field to an existing Java source file

```
field string --fieldName --class --column
```

--fieldName

The name of the field to add; default: *'NULL'* (mandatory)

--class

The name of the class to receive this field; default if option not present: *'*'* (mandatory)

--notNull

Whether this value cannot be null; default if option present: *'true'*; default if option not present: *'false'*

--nullRequired

Whether this value must be null; default if option present: *'true'*; default if option not present: *'false'*

--sizeMin

The minimum string length; default: *'NULL'*

--sizeMax

The maximum string length; default: *'NULL'*

--regexp

The required regular expression pattern; default: *'NULL'*

--column

The JPA @Column name; default: *'NULL'* (mandatory)

--value

Inserts an optional Spring @Value annotation with the given content; default: *'NULL'*

--comment

An optional comment for JavaDocs; default: *'NULL'*

--transient

Indicates to mark the field as transient; default if option present: *'true'*; default if option not present: *'false'*

--unique

Indicates whether to mark the field with a unique constraint; default if option present: *'true'*; default if option not present: *'false'*

--permitReservedWords

Indicates whether reserved words are ignored by Roo; default if option present: *'true'*; default if option not present: *'false'*

--lob

Indicates that this field is a Large Object; default if option present: *'true'*; default if option not present: *'false'*

Finder Commands

Finder Commands are contained in `org.springframework.roo.addon.finder.addon.FinderCommands`.

finder add

Install a finder in the given target (must be an entity)

```
finder add --class --name
```

--class

The entity for which the finders are generated; default if option not present: '*' (mandatory)

--name

The finder string defined as a Spring Data query; default: 'NULL' (mandatory)

Help Commands

Help Commands are contained in `org.springframework.roo.felix.help.HelpCommands`.

help

Shows system help

```
help
```

--command

Command name to provide help for; default: *'NULL'*

reference guide

Writes the reference guide XML fragments (in DocBook format) into the current working directory

```
reference guide
```

This command does not accept any options.

Hint Commands

Hint Commands are contained in `org.springframework.roo.classpath.operations.HintCommands`.

hint

Provides step-by-step hints and context-sensitive guidance

```
hint
```

--topic

The topic for which advice should be provided

I18n Commands

I18n Commands are contained in `org.springframework.roo.addon.web.mvc.i18n.I18nCommands`.

web mvc language

Install new internationalization bundle for MVC views.

```
web mvc language --code --module
```

--code

The language code for the desired bundle; default: *NULL* (mandatory)

--module

The application module where to install message bundles; default if option not present: '.' (mandatory)

Integration Test Commands

Integration Test Commands are contained in `org.springframework.roo.addon.test.addon.integration.IntegrationTestCommands`.

test integration

Creates a new integration test for the specified entity

```
test integration
```

--entity

The name of the entity to create an integration test for; default if option not present: '*'

--permitReservedWords

Indicates whether reserved words are ignored by Roo; default if option present: 'true'; default if option not present: 'false'

--transactional

Indicates whether the created test cases should be run withing a Spring transaction; default: 'true'

J Line Shell Component

J Line Shell Component are contained in
`org.springframework.roo.shell.jline.osgi.JLineShellComponent`.

`*/ ~~`

End of block comment

`*/`

This command does not accept any options.

`/* ~~`

Start of block comment

`/*`

This command does not accept any options.

`~~`

Inline comment markers (start of line only)

This command does not accept any options.

date

Displays the local date and time

```
date
```

This command does not accept any options.

script

Parses the specified resource file and executes its commands

```
script --file
```

--file

The file to locate and execute; default: *'NULL'* (mandatory)

--lineNumbers

Display line numbers when executing the script; default if option present: 'true'; default if option not present: 'false'

system properties

Shows the shell's properties

```
system properties
```

This command does not accept any options.

version

Displays shell version

```
version
```

--[default]

Special version flags; default: '*NULL*'

Jpa Commands

Jpa Commands are contained in `org.springframework.roo.addon.jpa.addon.JpaCommands`.

embeddable

Creates a new Java class source file with the JPA `@Embeddable` annotation in `SRC_MAIN_JAVA`

```
embeddable --class
```

--class

The name of the class to create; default: *'NULL'* (mandatory)

--serializable

Whether the generated class should implement `java.io.Serializable`; default if option present: *'true'*; default if option not present: *'false'*

--permitReservedWords

Indicates whether reserved words are ignored by Roo; default if option present: *'true'*; default if option not present: *'false'*

entity jpa

Creates a new JPA persistent entity in `SRC_MAIN_JAVA`

```
entity jpa --class --table --identifierColumn --versionField --versionColumn  
--versionType --sequenceName --identifierStrategy
```

--class

Name of the entity to create; default: *'NULL'* (mandatory)

--extends

The superclass (defaults to `java.lang.Object`); default if option not present: *'java.lang.Object'*

--implements

The interface to implement; default: *'NULL'*

--abstract

Whether the generated class should be marked as abstract; default if option present: *'true'*; default if option not present: *'false'*

--testAutomatically

Create automatic integration tests for this entity; default if option present: 'true'; default if option not present: 'false'

--table

The JPA table name to use for this entity; default: 'NULL' (mandatory)

--schema

The JPA table schema name to use for this entity; default: 'NULL'

--catalog

The JPA table catalog name to use for this entity; default: 'NULL'

--identifierField

The JPA identifier field name to use for this entity; default: 'NULL'

--identifierColumn

The JPA identifier field column to use for this entity; default: 'NULL' (mandatory)

--identifierType

The data type that will be used for the JPA identifier field (defaults to java.lang.Long); default: 'java.lang.Long'

--versionField

The JPA version field name to use for this entity; default: 'NULL' (mandatory)

--versionColumn

The JPA version field column to use for this entity; default: 'NULL' (mandatory)

--versionType

The data type that will be used for the JPA version field (defaults to java.lang.Integer); default if option not present: 'java.lang.Integer' (mandatory)

--inheritanceType

The JPA @Inheritance value (apply to base class); default: 'NULL'

--mappedSuperclass

Apply @MappedSuperclass for this entity; default if option present: 'true'; default if option not present: 'false'

--equals

Whether the generated class should implement equals and hashCode methods; default if option present: 'true'; default if option not present: 'false'

--serializable

Whether the generated class should implement java.io.Serializable; default if option present: 'true'; default if option not present: 'false'

--permitReservedWords

Indicates whether reserved words are ignored by Roo; default if option present: 'true'; default if option not present: 'false'

--entityName

The name used to refer to the entity in queries; default: 'NULL'

--sequenceName

The name of the sequence for incrementing sequence-driven primary keys; default: 'NULL' (mandatory)

--identifierStrategy

The generation value strategy to be used; default if option present: 'AUTO' (mandatory)

--readOnly

Whether the generated entity should be used for read operations only; default if option present: 'true'; default if option not present: 'false'

jpa setup

Install or updates a JPA persistence provider in your project

```
jpa setup --provider --database --module
```

--provider

The persistence provider to support; default: 'NULL' (mandatory)

--database

The database to support; default: 'NULL' (mandatory)

--module

The application module where to install the persistence; default if option not present: '.' (mandatory)

--jndiDataSource

The JNDI datasource to use; default: 'NULL'

--hostName

The host name to use; default: 'NULL'

--databaseName

The database name to use; default: *'NULL'*

--userName

The username to use; default: *'NULL'*

--password

The password to use; default: *'NULL'*

Maven Commands

Maven Commands are contained in `org.springframework.roo.project.MavenCommands`.

dependency add

Adds a new dependency to the Maven project object model (POM)

```
dependency add --groupId --artifactId --version
```

--groupId

The group ID of the dependency; default: *'NULL'* (mandatory)

--artifactId

The artifact ID of the dependency; default: *'NULL'* (mandatory)

--version

The version of the dependency; default: *'NULL'* (mandatory)

--classifier

The classifier of the dependency; default: *'NULL'*

--scope

The scope of the dependency; default: *'NULL'*

dependency remove

Removes an existing dependency from the Maven project object model (POM)

```
dependency remove --groupId --artifactId --version
```

--groupId

The group ID of the dependency; default: *'NULL'* (mandatory)

--artifactId

The artifact ID of the dependency; default: *'NULL'* (mandatory)

--version

The version of the dependency; default: *'NULL'* (mandatory)

--classifier

The classifier of the dependency; default: *'NULL'*

maven repository add

Adds a new repository to the Maven project object model (POM)

```
maven repository add --id --url
```

--id

The ID of the repository; default: *'NULL'* (mandatory)

--name

The name of the repository; default: *'NULL'*

--url

The URL of the repository; default: *'NULL'* (mandatory)

maven repository remove

Removes an existing repository from the Maven project object model (POM)

```
maven repository remove --id --url
```

--id

The ID of the repository; default: *'NULL'* (mandatory)

--url

The URL of the repository; default: *'NULL'* (mandatory)

module create

Creates a new Maven module

```
module create --moduleName --parent
```

--moduleName

The name of the module; default: *'NULL'* (mandatory)

--parent

The parent module name. By default is the current module; default: *'NULL'* (mandatory)

--packaging

The Maven packaging of this module; default if option not present: 'jar'

--artifactId

The artifact ID of this module (defaults to moduleName if not specified); default: *'NULL'*

module focus

Changes focus to a different project module

```
module focus --moduleName
```

--moduleName

The module to focus on; default: *'NULL'* (mandatory)

perform assembly

Executes the assembly goal via Maven

```
perform assembly
```

This command does not accept any options.

perform clean

Executes a full clean (including Eclipse files) via Maven

```
perform clean
```

This command does not accept any options.

perform command

Executes a user-specified Maven command

```
perform command --mavenCommand
```

--mavenCommand

User-specified Maven command (eg test:test); default: 'NULL' (mandatory)

perform eclipse

Sets up Eclipse configuration via Maven (only necessary if you have not installed the m2eclipse plugin in Eclipse)

```
perform eclipse
```

This command does not accept any options.

perform package

Packages the application using Maven, but does not execute any tests

```
perform package
```

This command does not accept any options.

perform tests

Executes the tests via Maven

```
perform tests
```

This command does not accept any options.

Metadata Commands

Metadata Commands are contained in `org.springframework.roo.classpath.MetadataCommands`.

metadata cache

Shows detailed metadata for the indicated type

```
metadata cache --maximumCapacity
```

--maximumCapacity

The maximum number of metadata items to cache; default: *'NULL'* (mandatory)

metadata for id

Shows detailed information about the metadata item

```
metadata for id --metadataId
```

--metadataId

The metadata ID (should start with MID:); default: *'NULL'* (mandatory)

metadata for module

Shows the ProjectMetadata for the indicated project module

```
metadata for module
```

--module

The module for which to retrieve the metadata (defaults to the focused module); default: *'NULL'*

metadata for type

Shows detailed metadata for the indicated type

```
metadata for type --type
```

--type

The Java type for which to display metadata; default: '*NULL*' (mandatory)

metadata status

Shows metadata statistics

```
metadata status
```

This command does not accept any options.

metadata trace

Traces metadata event delivery notifications

```
metadata trace --level
```

--level

The verbosity of notifications (0=none, 1=some, 2=all); default: '*NULL*' (mandatory)

Obr Add On Commands

Obr Add On Commands are contained in
org.springframework.roo.obr.addon.search.ObrAddOnCommands.

addon info bundle

Provide information about a specific Spring Roo Add-on

```
addon info bundle --bundleSymbolicName
```

--bundleSymbolicName

The bundle symbolic name for the add-on of interest; default: '*NULL*' (mandatory)

addon install bundle

Install Spring Roo Add-on

```
addon install bundle --bundleSymbolicName
```

--bundleSymbolicName

The bundle symbolic name for the add-on of interest; default: '*NULL*' (mandatory)

addon install url

Install Spring Roo Add-on using url

```
addon install url --url
```

--url

The url for the add-on of interest; default: '*NULL*' (mandatory)

addon list

List all installed addons

```
addon list
```


This command does not accept any options.

addon remove

Remove Spring Roo Add-on

```
addon remove --bundleSymbolicName
```

--bundleSymbolicName

The bundle symbolic name for the add-on of interest; default: *'NULL'* (mandatory)

addon search

Search all known Spring Roo Add-ons

```
addon search --requiresCommand
```

--requiresCommand

Only display add-ons in search results that offer this command; default: *'NULL'* (mandatory)

Obr Repository Commands

Obr Repository Commands are contained in `org.springframework.roo.obr.addon.search.ObrRepositoryCommands`.

addon repository add

Adds a new OBR Repository to ROO Shell

```
addon repository add --url
```

--url

URL file that defines repository. Ex: 'http://localhost/repo/index.xml'; default: 'NULL' (mandatory)

addon repository introspect

Introspects all installed OBR Repositories and list all their addons

```
addon repository introspect
```

This command does not accept any options.

addon repository list

Lists existing OBR Repositories

```
addon repository list
```

This command does not accept any options.

addon repository remove

Removes an existing OBR Repository from ROO Shell

```
addon repository remove --url
```

--url

URL file that defines repository. Ex: 'http://localhost/repo/index.xml'; default: 'NULL' (mandatory)

Os Commands

Os Commands are contained in `org.springframework.roo.addon.oscommands.OsCommands`.

! ~

Allows execution of operating system (OS) commands.

- ! -

--command

The command to execute; default: "

Pgp Commands

Pgp Commands are contained in `org.springframework.roo.felix.pgp.PgpCommands`.

pgp automatic trust

Indicates to automatically trust all keys encountered until the command is invoked again

```
pgp automatic trust
```

This command does not accept any options.

pgp key view

Downloads a remote key and displays it to the user (does not change any trusts)

```
pgp key view --keyId
```

--keyId

The key ID to view (eg 00B5050F or 0x00B5050F); default: '*NULL*' (mandatory)

pgp list trusted keys

Lists the keys you currently trust and have not been revoked at the time last downloaded from a public key server

```
pgp list trusted keys
```

This command does not accept any options.

pgp refresh all

Refreshes all keys from public key servers

```
pgp refresh all
```

This command does not accept any options.

pgp status

Displays the status of the PGP environment

```
pgp status
```

This command does not accept any options.

pgp trust

Grants trust to a particular key ID

```
pgp trust --keyId
```

--keyId

The key ID to trust (eg 00B5050F or 0x00B5050F); default: *'NULL'* (mandatory)

pgp untrust

Revokes your trust for a particular key ID

```
pgp untrust --keyId
```

--keyId

The key ID to remove trust from (eg 00B5050F or 0x00B5050F); default: *'NULL'* (mandatory)

Process Manager Diagnostics Listener

Process Manager Diagnostics Listener are contained in
org.springframework.roo.process.manager.internal.ProcessManagerDiagnosticsListener.

process manager debug

Indicates if process manager debugging is desired

```
process manager debug
```

--enabled

Activates debug mode; default: 'true'

Project Commands

Project Commands are contained in `org.springframework.roo.project.ProjectCommands`.

development mode

Switches the system into development mode (greater diagnostic information)

```
development mode
```

--enabled

Activates development mode; default: 'true'

project scan now

Perform a manual file system scan

```
project scan now
```

This command does not accept any options.

project scan speed

Changes the file system scanning speed

```
project scan speed --ms
```

--ms

The number of milliseconds between each scan; default: 'NULL' (mandatory)

project scan status

Display file system scanning information

```
project scan status
```

This command does not accept any options.

project setup

Creates a new Maven project

```
project setup --topLevelPackage
```

--topLevelPackage

The uppermost package name (this becomes the <groupId> in Maven and also the '~' value when using Roo's shell); default: *'NULL'* (mandatory)

--projectName

The name of the project (last segment of package name used as default); default: *'NULL'*

--multimodule

Option to use a multimodule architecture; default if option present: *'STANDARD'*

--java

Forces a particular major version of Java to be used (DEFAULT: 8); default: *'NULL'*

--packaging

The Maven packaging of this project; default if option not present: *'jar'*

Prop File Commands

Prop File Commands are contained in `org.springframework.roo.addon.propfiles.PropFileCommands`.

property add

Adds or updates a particular property from application config properties file.

```
property add --key --value --module
```

--key

The property key that should be changed; default: *'NULL'* (mandatory)

--value

The new value for this property key; default: *'NULL'* (mandatory)

--module

Module where property will be added; default if option not present: *'.'* (mandatory)

property list

List all properties from application config properties file.

```
property list --module
```

--module

Module which properties will be listed; default if option not present: *'.'* (mandatory)

property remove

Removes a particular property from application config properties file.

```
property remove --key --module
```

--key

The property key that should be removed; default: *'NULL'* (mandatory)

--module

Module where property will be removed; default if option not present: *'.'* (mandatory)

Push In Commands

Push In Commands are contained in `org.springframework.roo.addon.pushin.PushInCommands`.

push-in

Push-in all methods, fields, annotations, imports, extends, etc.. declared on ITDs to its .java files. You could specify `--all` parameter to apply push-in on every component of generated project, or you could define package, class or method where wants to apply push-in.

```
push-in
```

--all

Parameter that indicates if push-in process should be applied to entire project. If specified, all the other parameters will be unavailable. It doesn't allow any value.

--package

JavaPackage with the specified package where developers wants to make push-in; default: *'NULL'*

--class

JavaType with the specified class where developer wants to make push-in; default: *'NULL'*

--method

String with the specified name of the method that developer wants to push-in. You could use a Regular Expression to make push-in of more than one method on the same execution.; default: *'NULL'*

Repository Jpa Commands

Repository Jpa Commands are contained in
org.springframework.roo.addon.layers.repository.jpa.addon.RepositoryJpaCommands.

repository jpa

Generates new Spring Data repository for specified entity.

```
repository jpa --interface --package
```

--all

Indicates if developer wants to generate repositories for every entity of current project ; default if option present: 'true'; default if option not present: 'false'

--interface

The java Spring Data repository to generate.; default: '*NULL*' (mandatory)

--entity

The domain entity this repository should expose; default: '*NULL*'

--package

The package where repositories will be generated; default: '*NULL*' (mandatory)

Security Commands

Security Commands are contained in
org.springframework.roo.addon.security.addon.security.SecurityCommands.

security setup

Install Spring Security into your project

```
security setup --module
```

--module

The application module where to install the persistence; default if option not present: ''
(mandatory)

Service Commands

Service Commands are contained in `org.springframework.roo.addon.layers.service.addon.ServiceCommands`.

service

Creates new service interface and its implementation.

```
service --repository --interface --apiPackage --implPackage
```

--all

Indicates if developer wants to generate service interfaces and their implementations for every entity of current project ; default if option present: 'true'; default if option not present: 'false'

--entity

The domain entity this service should expose; default: 'NULL'

--repository

The repository this service should expose; default: 'NULL' (mandatory)

--interface

The service interface to be generated; default: 'NULL' (mandatory)

--class

The service implementation to be generated; default: 'NULL'

--apiPackage

The java interface package; default: 'NULL' (mandatory)

--implPackage

The java package of the implementation classes for the interfaces; default: 'NULL' (mandatory)

Settings Commands

Settings Commands are contained in `org.springframework.roo.settings.SettingsCommands`.

settings add

Adds or updates a particular setting

```
settings add --name --value
```

--name

The setting name that should be changed; default: *'NULL'* (mandatory)

--value

The new value for this settings name; default: *'NULL'* (mandatory)

settings list

Lists all settings added into configuration

```
settings list
```

This command does not accept any options.

settings remove

Removes an specific setting from configuration

```
settings remove --name
```

--name

The settings name that should be removed; default: *'NULL'* (mandatory)

Unit Test Commands

Unit Test Commands are contained in `org.springframework.roo.addon.test.addon.unit.UnitTestCommands`.

test unit

Creates a unit test class for the specified class

```
test unit --class
```

--class

The name of the project class which this unit test class is targeting; default: '*NULL*' (mandatory)

--permitReservedWords

Indicates whether reserved words are ignored by Roo; default if option present: 'true'; default if option not present: 'false'

View Commands

View Commands are contained in `org.springframework.roo.addon.web.mvc.views.ViewCommands`.

web mvc templates setup

Includes view generation templates on current project. Will allow developers to customize view generation.

```
web mvc templates setup --type
```

--type

View identifier of templates you want to install. Only installed views are available.; default: 'NULL' (mandatory)

web mvc view setup

Includes all necessary resources of provided responseType on generated project

```
web mvc view setup --type --module
```

--type

View identifier you want to install. Install your necessary views before to be used on controller generation command; default: 'NULL' (mandatory)

--module

The application module where to install views; default if option not present: '.' (mandatory)

Web Finder Commands

Web Finder Commands are contained in `org.springframework.roo.addon.web.mvc.controller.addon.finder.WebFinderCommands`.

web mvc finder

Adds `@RooWebFinder` annotation to MVC controller type

```
web mvc finder --controller --responseType
```

--controller

The controller java type to apply this annotation to.; default: *'NULL'* (mandatory)

--all

Indicates if developer wants to generate all finders of the service related to the controller; default if option present: *'true'*; default if option not present: *'false'*

--finder

Indicates the name of the finder to add into controller; default: *'NULL'*

--responseType

Indicates the responseType to be used by generated controller. Depending of the selected responseType, generated methods and views will vary.; default: *'NULL'* (mandatory)

6. Roo Resources

As an open source project, Spring Roo offers a large number of resources to assist the community learn, interact with one another and become more involved in the project. Below you'll find a short summary of the official project resources.

Spring Roo Project Home Page

The definitive source of information about Spring Roo is the [Spring Roo Home](http://spring.io) at <http://spring.io>.

That site provides a brief summary of Roo's main features and links to most of the other project resources. The project home page serves as a hub of information and is the best place to find up-to-date announcements about the project as well as links to articles, blogs and new documentation.

Please use this URI if you are referring other people to the Spring Roo project, as it is the main landing point for the project.

Downloads and Maven Repositories

You can always access the latest Spring Roo release ZIP by visiting Downloads section at [Spring Roo Home Page](#).

We publish all Roo modules to Maven Central, the default repository from which Maven will download the Spring Roo artifacts automatically.

StackOverFlow

Because Roo is an official top-level Spring project, of course you'll find there is a dedicated "Spring Roo" tag at Stack Overflow for all your questions, comments and experiences.

If you have any question about Spring Roo project and its functionalities, you can check and ask a questions at [Spring Roo tagged questions at Stack Overflow](#). We monitor stackoverflow.com for questions tagged with spring-roo.

<http://forum.springsource.org> is now a read-only archive. All commenting, posting, registration services have been turned off.

The Roo project does not have a "mailing list" or "newsgroup" as you might be familiar with from other open source projects, although [commercial support](#) options are available.

Extensive search facilities are provided on the community forums, and the Roo developers routinely answer user questions. One excellent way of contributing to the Roo project is to simply keep an eye on the forum messages and help other people. Even recommendations along the lines of, "I don't know how to do what you're trying to do, but we usually tackle the problem this way instead...." are very helpful to other community members.

When you ask a question on the forum, it's highly recommended you include a small Roo [sample script](#) that can be used to reproduce your problem. If that's infeasible, using Roo's "[backup](#)" command is another alternative and you can attach the resulting ZIP file to your post. Other tips include always specifying the version of Roo that you're running (as can be obtained from the "[version](#)" command), and if you're having trouble with IDE integration, the exact version of the IDE you are using (and, if an Eclipse-based IDE, the version of [AspectJ Development Tools](#) in use). Another good source of advice on how to ask questions on the forum can be found in Eric Raymond's often-cited essay, "[How to Ask Smart Questions](#)".

If you believe you have found a bug or are experiencing an issue, it is recommended you first log a message on the forum. This allows other experienced users to comment on whether it appears there is a problem with Roo or perhaps just needs to be used a different way. Someone will usually offer a solution or recommend you log a bug report (usually by saying "please log this in Jira"). When you do log a bug report, please ensure you link to the fully-qualified URI to the forum post. That way the developer who attempts to solve your bug will have background information. Please also post the issue tracking link back in thread you started on the forum, as it will help other people cross-reference the two systems.

Twitter

Roo Hash Code (please include in your tweets, and also follow for low-volume announcements): [#SpringRoo](#)

If you use Twitter, you're encouraged to follow @SpringRoo. Also please use @SpringRoo in your tweets so everyone can easily see them.

The Roo team also uses and monitors tweets that include #SpringRoo, so if you're tweeting about Roo, please remember to include #SpringRoo somewhere in the tweet. If you like Roo or have found it helpful on a project, please tweet about it and help spread the word!

Follow the core Roo development team for interesting Roo news and progress (higher volume than just following @SpringRoo, but only a few Tweets per week): [@disid_corp](#), [@juanCaFX](#), [@enrique_ruiz_](#).

Many people who use Roo also use Twitter, including the core Roo development team. If you're a Twitter user, you're welcome to follow the Roo development team (using the Twitter IDs above) to receive up-to-the-minute Tweets on Roo activities, usage and events.

We do request that you use the [StackOverflow](#) if you have a question or issue with Roo, as 140 characters doesn't allow us to provide in-depth technical support or provide a growing archive of historical answers that people can search against.

Issue Tracking

Web: <https://jira.spring.io/browse/ROO/>

Spring projects use Atlassian Jira for tracking bugs, improvements, feature requests and tasks. Roo uses a public Jira instance you're welcome to use in order to log issues, watch existing issues, vote for existing issues and review the changes made between particular versions.

As discussed in the [StackOverflow](#) section, we ask that you refrain from logging bug reports until you've first discussed them on stackoverflow. This allows others to comment on whether a bug actually exists. When logging an issue in Jira, there is a field explicitly provided so you can link the forum discussion to the Jira issue.

Please note that every commit into the Roo [source repository](#) will be prefixed with a particular Jira issue number. All Jira issue numbers for the Roo project commence with "ROO-", providing you an easy way to determine the rationale of any change.

Because open source projects receive numerous enhancement requests, we generally prioritise enhancements that have patches included, are quick to complete or those which have received a large number of votes. You can vote for a particular issue by logging into Jira (it's fast, easy and free to create an account) and click the "vote" link against any issue. Similarly you can monitor the progress on any issue you're interested in by clicking "watch".

Enhancement requests are easier to complete (and therefore more probable to be actioned) if they

represent fine-grained units of work that include as much detail as possible. Enhancement requests should describe a specific use case or user story that is trying to be achieved. It is usually helpful to provide a Roo [sample script](#) that can be used to explain the issue. You should also consider whether a particular enhancement is likely to appeal to most Roo users, and if not, whether perhaps writing it as an [add-on](#) would be a good alternative.

Source Repository

Read repository: <https://github.com/spring-projects/spring-roo.git>

The Git source control system is currently used by Roo for mainline development.

Historical releases of Roo can be accessed by browsing the tags branches within our Git repository. The mainline development of Roo occurs on the "master" branch.

"gh-pages" branch is used to build and publish Spring Roo's project page site based on Jekyll and GitHub Pages.

To detailed information about how to check out and build Roo from Subversion, please refer to the [Development Processes](#) chapter.

Commercial Products and Services

Web: <http://www.disid.com/en/>

DISID Corporation employs the Roo development team and offers a wide range of products and professional services around Roo and the technologies which Roo enables. Available professional services include software factory, geographic information systems, web application development, mobile application development, training, consulting and mentoring. Please visit the above URI to learn more about DISID products and services.

Web: <http://spring.io/>

Pivotal Software offers a wide range of products and professional services around Roo and the technologies which Roo enables. Available professional services include training, consulting, design reviews and mentoring, with products including service level agreement (SLA) backed support subscriptions, certified builds, indemnification and integration with various commercial products. Please visit the above URI to learn more about SpringSource products and services and how these can add value to your build-run-manage application lifecycle.

Other

Please let us know if you believe it would be helpful to list any other resources in this documentation.