

Spring Boot MongoDB

Filed Under: [MongoDB](#)

Shubham - 26 Comments

[Home](#) » [Spring](#) » Spring Boot MongoDB

MongoDB Tutorials

- 1. MongoDB Tutorial
- 2. Install MongoDB on Windows
- 3. Install MongoDB on Mac OS
- 4. Install MongoDB on Linux
- 5. Install MongoDB on CentOS
- 6. MongoDB Insert
- 7. MongoDB Update
- 8. MongoDB Find
- 9. MongoDB FindOne
- 10. MongoDB FindAndModify
- 11. MongoDB Sort
- 12. MongoDB Bulk Insert
- 13. MongoDB Distinct
- 14. MongoDB Map Reduce
- 15. MongoDB Upsert
- 16. MongoDB Remove
- 17. MongoDB Authentication
- 18. MongoDB Exists
- 19. MongoDB Java CRUD
- 20. MongoDB Java Servlet
- 21. MongoDB Spring Example
- 22. MongoDB Spring Boot
- 23. Spring Data MongoDB

Welcome to Spring Boot MongoDB example. Spring Boot is the easiest way to spin a spring project quickly and MongoDB is the most popular NoSQL database. Let's see how to integrate spring with MongoDB database.

Table of Contents	hi [d] e
1 Spring Boot	
MongoDB	
1.1 Spring	



Spring Framework

- 1. Spring Framework
- 2. Spring 5
- 3. Spring WebFlux

JournalDev

< Previous Next >



- 6. Spring IoC and Bean
- 7. Spring Bean Life Cycle
- 8. Spring REST
- 9. Spring REST XML
- 10. Spring RestTemplate
- 11. Spring AOP
- 12. Spring AOP Method Profiling
- 13. Spring Annotations
- 14. Spring @Autowired
- 15. Spring @RequestMapping
- 16. Spring MVC Example
- 17. Spring MVC Tutorial
- 18. Spring MVC Exception Handling
- 19. Spring MVC Validator
- 20. Spring MVC Interceptor
- 21. Spring MVC File Upload
- 22. Spring MVC i18n
- 23. Spring MVC Hibernate MySQL
- 24. Spring ORM
- 25. Spring ORM JPA
- 26. Spring Data JPA
- 27. Spring Transaction
- 28. Spring JdbcTemplate
- 29. Spring Security Overview
- 30. Spring Security Example Tutorial
- 31. Spring Security UserDetailsService
- 32. Spring MVC Login Logout
- 33. Spring Security Roles
- 34. Spring Boot Tutorial
- 35. Spring Boot Components
- 36. Spring Boot CLI Hello World
- 37. Spring Boot Initializr Web
- 38. Spring Boot Initializr IDE
- 39. Spring Boot Initializr CLI
- 40. Spring Boot Initializr Tools
- 41. Spring Boot MongoDB
- 42. Spring Boot Redis Cache
- 43. Spring Boot Interview Questions

- 1.4 Spring Boot MongoDB APIs
- 2 Spring Data MongoDB – MongoRepository
- 2.1 Defining MongoDB properties
- 2.2 Defining the Spring Controller
- 2.3 Defining the APIs
 - 2.3.1 Getting all users
 - 2.3.2 Getting a user by ID
 - 2.3.3 Adding a new User
 - 2.3.4 Getting



JournalDev

< Previous Next >



- 47. Spring RabbitMQ
- 48. Spring AMQP RabbitMQ
- 49. Apache ActiveMQ
- 50. Spring ActiveMQ Tutorial
- 51. Spring ActiveMQ Example
- 52. Spring JDBC
- 53. Spring DataSource JNDI
- 54. Spring Hibernate
- 55. Spring Primefaces JPA
- 56. Spring Primefaces MongoDB
- 57. Spring Primefaces Hibernate
- 58. SpringRoo Primefaces Hibernate
- 59. Spring JSF
- 60. Spring JDF Hibernate
- 61. Spring Data MongoDB
- 62. Spring Interview Questions

3 Spring Data
MongoDB –
MongoTemplate
3.0.1
Defining
the DAL
interface
3.0.2
Impleme
nting the
DAL
interface
4 Spring Data
MongoDB Test
Run
5
MongoTemplate
vs
MongoReposito
ry
6 Spring Boot
MongoDB
Summary

Spring Boot MongoDB

We
API
Spr
Mo
dat



- Spring Data MongoDB
- Spring Boot

There are two approaches through which we can connect to MongoDB database – `MongoRepository` and `MongoTemplate`. We will try to establish what one API offers over another and when should you choose any one of them for your use-case. We will make use of `Spring Initializr` tool for quickly setting up the project. So, let's get started.

Spring Boot MongoDB Project Setup

We will start off by creating a new Spring Boot application using the `Spring Initializr` tool. This will help us quickly set up the project and use the required dependencies.

The Orion Multi-Purpose Crew Vehicle is a US-European spacecraft intended to carry a crew of four astronauts to destinations at or beyond low Earth orbit

JournalDev

< Previous Next >



The screenshot shows the Spring Initializr interface. On the left, there's a 'Project Metadata' section with fields for Group, Artifact, Version, Description, and Archetype. On the right, there's a 'Dependencies' section with a search bar and a list of selected dependencies. A large green 'Generate Project' button is at the bottom.

Download the project and unzip it. Then import it into your favorite IDE – Eclipse or IntelliJ IDEA.

Maven Dependencies

Though we already completed the setup with the tool, if you want to set it up manually, we use Maven build system for this project and here are the dependencies we used:

```
<?xml vers
<project x
  xs
  <m
    <g
    <a
    <v
    <p
```



JournalDev

[◀ Previous](#) [Next ▶](#)

Make sure to use stable version for **Spring Boot** from the maven central.

Spring Boot MongoDB Model Class

We have a simple model class

`User.java .`

```
pu
}
pu
```



We will have following functionalities and Database interactions in our app.

- Get all users
- Get a user with ID
- Get user settings
- Get a particular key from the Map
- Add/Update user setting

Spring Data MongoDB

MongoRepository

Now we will use Spring Data MongoDB repository to access our data.

Spring Data MongoRepository provide us common functionalities that we can easily

plu

Let

Rep

inte



JournalDev

[◀ Previous](#) [Next ▶](#)

Defining MongoDB properties

Before we lay out our controller, it is important that we make a connection with a local instance of MongoDB. We will use Spring Boot properties to do this.

```
#Local Mongc  
spring.data.  
spring.data.  
spring.data.  
spring.data.  
spring.data.  
spring.data.  
spring.data.
```

```
# App config  
server.port=  
spring.appli  
server.conte
```

So, the app will run on port 8102 and connect to a local mongoDB instance with

pro

cre

hav

inst

aut

ena

just

firs

con



The Orion Multi-Purpose Crew Vehicle is a US-European spacecraft intended to carry a crew of four astronauts to destinations at or beyond low Earth orbit

Let us finally move to making our Controller class.

```
package com.  
  
import java.  
  
import org.s  
  
import com.j  
import com.j  
  
@RestController  
@RequestMapping  
public class  
  
    private  
    private  
    public  
}  
}
```

We just Autowired the repository interface dependency and we next

De
the

For
fun
me
now
API

The Orion Multi-Purpose Crew Vehicle is a US-European spacecraft intended to carry a crew of four astronauts to destinations at or beyond low Earth orbit

will internally use Spring Data MongoRepository API. Notice that we do not have to write any database interaction code in the interface as Spring Data does it all for us.

Getting all users

```
@RequestMapping  
public List<  
    LOG.  
    retu  
}
```

`findAll()` is just a method which Spring Data MongoRepository provides internally.

Getting a user by ID

Now, let us get a specific user with an ID:



by an ID.

Adding a new User

We will be adding a new user in the function below.

```
@RequestMapp  
public User  
    LOG.  
retu  
}
```

Getting User settings

Now that we have added sample data into the DB, let's try to extract some part of it.

```
@RequestMapp  
public Objec  
    User  
if (  
    } el  
    }  
}
```

Ge
pa
sel

The Orion Multi-Purpose Crew Vehicle is a US-European spacecraft intended to carry a crew of four astronauts to destinations at or beyond low Earth orbit

Notice in the above query, we got the user object, then extracted the complete Setting map (which could have contained 1000s of objects) and finally got our own value. This is a downside for Spring Data query when we use it as straight API.

Adding a new User setting

Let's try to add some data to an existing user:

```
@RequestMapp  
public String  
    User  
    if (
```



```
} el  
}  
}
```



JournalDev

[◀ Previous](#) [Next ▶](#)

THIS IS THE EASE

Spring Data
MongoReposit
ory API offers us
but it also has
some downsides.
We will elaborate
this when we have
defined the
MongoTemplat
e version as well.
Let's get started
with that too.

Spring Data MongoDB

MongoTemplate

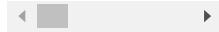
We will be
defining the
MongoTemplate
database queries
here. With
MongoTemplate,
you will see that
we have much
more granular
control over what
we query and
what data is
included in the
results.

De
DA

To
cor
dat
lay
by
int
wo

The Orion Multi-Purpose Crew Vehicle is a US-European spacecraft intended
to carry a crew of four astronauts to destinations at or beyond low Earth orbit

```
package com.  
import java.  
import com.j  
public inter  
    List  
    User  
    User  
    Obje  
    Stri  
    Stri  
}
```



Implementing the DAL interface

Let's move on and define these methods.

```
pu  
}  
@O  
pu
```



JournalDev

[◀ Previous](#) [Next ▶](#)

The method implementations in above class are using MongoTemplate dependency.

See how

```
getUserById(  
...)
```

method gets the user. We construct a query and passed required parameters.

What will interests you more is the `getUserSetting` query. Let us understand what happened above:

- We constructed queries with criteria to check equality.
- The `include` method includes the field names which the result should include when it



JournalDev

[◀ Previous](#) [Next ▶](#)

```
return empty  
data meaning  
the required  
key wasn't  
found. This  
saves from  
even fetching  
the User object  
at all if the  
required key  
was not  
present
```

Spring Data MongoDB Test Run

We can run this app simply by using a single command:

```
mvn spring-t
```

Once the app is running, we can try saving a new user by using this API:

```
https://locat
```

As this will be a
PO
will
JSO
wel

The Orion Multi-Purpose Crew Vehicle is a US-European spacecraft intended to carry a crew of four astronauts to destinations at or beyond low Earth orbit

JournalDev

[◀ Previous](#) [Next ▶](#)


Mongo response itself, we will get something like:

```
{
  "userId": "S
  "name": "S
  "creationD
  "userSetti
  "bike" :
}
}
```

You can get all users by using the API as GET request:

<https://localhost:8080/users>

We will get back something like:

```
[
  {
    "userId"
    "name": "S
    "creationD
    "userSetti
    "bike" :
```

The Orion Multi-Purpose Crew Vehicle is a US-European spacecraft intended to carry a crew of four astronauts to destinations at or beyond low Earth orbit

JournalDev

< Previous Next >



haven't hooked up MongoTemplate to be used. Below code snippet shows the changes required to use MongoTemplate for reading user settings.

```
//define Dat
private final MongoTemplate mongoTemplate;

//initialize
public UserC
    this
    this
}

//change met
@RequestMapping("User")
public Object getUser(
    User user
) {
    if (user != null)
        return user;
    else {
        return null;
    }
}

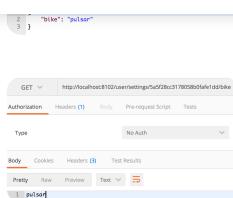
//change met
@RequestMapping("User")
public String
    @PathVariable("id")
    return user;
}
```

Res
and
to g
setti
any
Bel
sho
from
app

The Orion Multi-Purpose Crew Vehicle is a US-European spacecraft intended to carry a crew of four astronauts to destinations at or beyond low Earth orbit

JournalDev

< Previous Next >



MongoTemplate vs MongoRepository

- MongoTemplate provides a lot more control when it comes to querying data and what data to pull from database.
- Spring Data repositories provide us a convenient outlook on how to fetch data.
- MongoTemplate is database dependent. What this means is, with Spring Data repositories, you can easily switch to a different database altogether by simply using a different Spring Data



JournalDev

[◀ Previous](#) [Next ▶](#)

In this lesson, we looked at how MongoTemplate can provide us more control over Spring Data repositories but can also be a little complicated when deeper queries are involved. So, this is completely your call what to choose when you develop your idea. Feel free to leave comments below.

Download the source code from below link. Please make sure that you change the MongoDB credentials before running the provided app.

[Download Spring Boot MongoDB Example Project](#)



Comments

June 27, 2020 at
Sharath says:
Raj 7 am

I am
getting
below
error
when
runnin
g the
applic
ation.
Can
someo
ne
help!

20-06-
27
22:44:
46.903
INFO
9680

—
[localh
ost:27
017]
org.mo
ngodb.
driver.c
luster :



JournalDev

< Previous Next >



```
goSoc  
ketOpe  
nExcep  
tion:  
Excepti  
on  
openin  
g  
socket  
at  
com.m  
ongod  
b.conn  
ection.  
Socket  
Stream  
.open(  
Socket  
Stream  
.java:6  
3) ~  
[mong  
odb-  
driver-  
core-  
3.4.3.j  
ar:na]  
at  
com.m  
ongod  
b.conn  
ection.I  
nternal  
Stream
```



JournalDev

< Previous Next >



```
at  
com.m  
ongod  
b.conn  
ection.  
Defaul  
tServer  
Monito  
r$Serv  
erMoni  
torRun  
nable.r  
un(Def  
aultSer  
verMo  
nitor.ja  
va:113  
) ~  
[mong  
odb-  
driver-  
core-  
3.4.3.j  
ar:na]  
at  
java.b  
ase/ja  
va.lan  
g.Thre  
ad.run(  
Unkno  
wn  
Source  
)
```



JournalDev

< Previous Next >



ase/ja
va.net.
DualSt
ackPla
inSock
etImpl.
waitFo
rConne
ct(Nati
ve
Metho
d) ~
[na:na]
at
java.b
ase/ja
va.net.
DualSt
ackPla
inSock
etImpl.
socket
Conne
ct(Unk
nown
Source
) ~
[na:na]
at
java.b
ase/ja
va.net.
Abstra
ctPlain



JournalDev

< Previous Next >



```
Impl.c
onnect
ToAddr
ess(Un
known
Source
)
~[na:na]
at
java.b
ase/ja
va.net.
Abstra
ctPlain
Socket
Impl.c
onnect
(Unkno
wn
Source
)
~[na:na]
at
java.b
ase/ja
va.net.
PlainS
ocketl
mpl.co
nnect(
Unkno
wn
Source
)
~
```



JournalDev

< Previous Next >



```
java.b  
ase/ja  
va.net.  
Socket  
.conne  
ct(Unk  
nown  
Source  
) ~  
[na:na]  
at  
com.m  
ongod  
b.conn  
ection.  
Socket  
Stream  
Helper.  
initializ  
e(Sock  
etStrea  
mHelp  
er.java:  
57) ~  
[mong  
odb-  
driver-  
core-  
3.4.3.j  
ar:na]  
at  
com.m  
ongod  
b.conn
```



JournalDev

< Previous Next >



comm

on

frames

omitte

d Reply

Anjali says:March 12, 2020 at
2:55 am

Thank

you so

much!

Reply

**Reza says:**January 17, 2020 at
8:10 pm

@Enab

leMon

goRep

ositori

es is

require

d in

order

to

autowi

ring

works

Reply

joe says:

January 6, 2020 at

11:22 pm

hello,

does



JournalDev

< Previous Next >



s?Reply

Sripal says:

December
26, 2019 [Chain](#)
6:02 pmY
o
u
c
a
n
a
d
d
it
a
p
p
li
c
a
ti
o
n
.p
r
o
p
e
rt
i
e
s

JournalDev

[◀ Previous](#) [Next ▶](#)

boot
with
mongo
db and
just
follow
ed the
same
steps
as
given
in the
docum
ent
and I
also
went
throug
h the
comm
ents to
check
out if I
am
missin
g
anythi
ng,
howev
er
when I
hit
GET
API



JournalDev

< Previous Next >



```
SHA-1,  
userNa  
me='ro  
ot',  
source  
='admi  
n',  
passw  
ord=,  
mecha  
nismPr  
opertie  
s={}];  
nested  
excepti  
on    is  
com.m  
ongod  
b.Mon  
goSec  
urityEx  
ceptio  
n:  
Excepti  
on  
authen  
ticatin  
g  
Mongo  
Creden  
tial{me  
chanis  
m=SC  
RAM-
```



JournalDev

< Previous Next >



```
able to  
login  
to  
mongo  
shell  
and  
see the  
user is  
create  
d  
here  
are my  
mongo  
db  
propert  
ies I  
have  
define  
d  
spring.  
data.m  
ongod  
b.auth  
enticat  
ion-  
databa  
se=ad  
min  
spring.  
data.m  
ongod  
b.user  
name=  
root
```



JournalDev

< Previous Next >



```
data.m
ongod
b.port=
27018
spring.
data.m
ongod
b.host
=local
host
authen
ticatio
nMech
anism
=SCRA
M-
SHA-1
I also
see
below
applic
ation
logs
com.m
ongod
b.Mon
goCom
mandE
xceptio
n:
Comm
and
failed
with
```



JournalDev

< Previous Next >



The
full
respon
se is {
“ok” :
0.0,
“errms
g” :
“Authe
nticati
on
failed.”

,

“code”
: 18,
“codeN
ame” :
“Authe
nticati
onFail
ed” }

Can
anyon
e help
here?

[Reply](#)

Satya says:

November
20, 2019 at
10:41 am

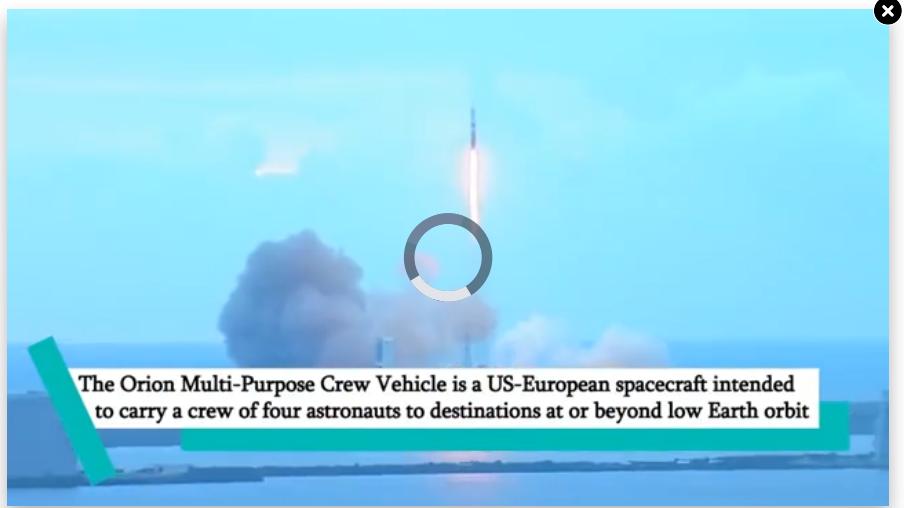
@
Y
a
t
h



JournalDev

[◀ Previous](#) [Next ▶](#)

e
a
b
o
v
e
b
y
c
r
e
a
t
i
n
g
u
s
e
r
a
n
d
p
w
d
a
s
b
e
l
o
w
.
>



JournalDev

[◀ Previous](#) [Next ▶](#)

```
e  
a  
t  
e  
U  
s  
e  
r(
```

{

u

s

e

r:

“r

o

o

t”

,

p

w

d

:

“r

o

o

t”

,

r

o

l

e

s

:

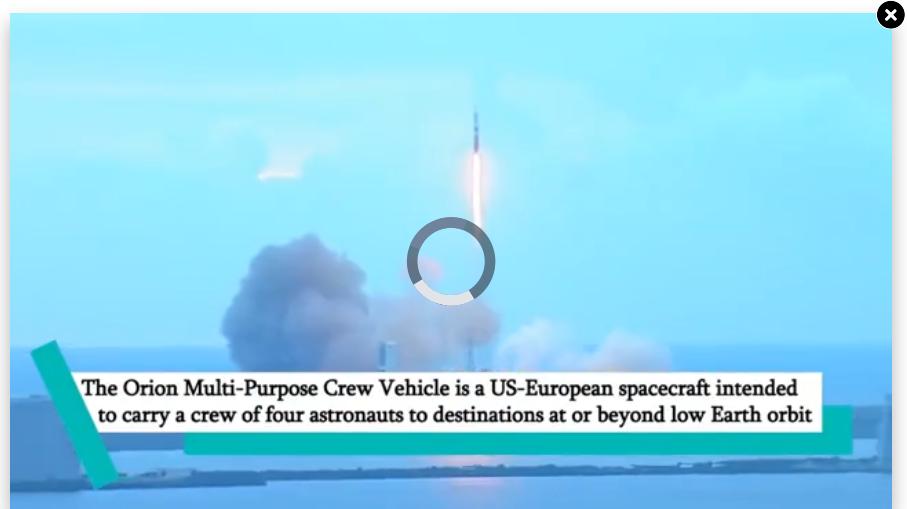
[



JournalDev

[◀ Previous](#) [Next ▶](#)

```
d  
m  
i  
n  
",  
d  
b  
:  
"  
a  
d  
m  
i  
n  
"  
},  
{  
r  
o  
l  
e  
:  
"r  
e  
a  
d  
A  
n  
y  
D  
a  
t  
a  
b
```



JournalDev

[◀ Previous](#) [Next ▶](#)

```
"r  
e  
a  
d  
W  
ri  
t  
e  
"]  
,  
{  
w  
:  
"  
m  
a  
j  
o  
ri  
t  
y  
"  
,w  
ti  
m  
e  
o  
u  
t:  
5  
0  
0  
0
```



JournalDev

< Previous Next >



ing
workin
g fine
but i
have
one
questi
on
i want
to run
project
spring
boot
with
mongo
Db but
how to
connec
tion
poolin
g like
maxim
um-
pool-
size,
max-
lifetim
e, idle-
timeou
t and if
i am
type
wrong
mongo



JournalDev

< Previous Next >



the
solutio
n

[Reply](#)

Shakthi says:
July 27, 2019 at
[Manain](#)

What
would
be the
mongo
db
config
uration
s if
mongo
db is
run in
a
docker
contai
ner?

[Reply](#)

Gokul says:
February 18, 2019
at 11:32 pm

UserRe
positor
y
findOn
e(Strin
g
UserID
)



JournalDev

[◀ Previous](#) [Next ▶](#)

a
s
e
c
h
e
c
k
t
h
e
s
p
r
i
n
g
v
e
r
s
i
o
n
y
o
u
a
r
e
u
s
i
n
g



JournalDev

[◀ Previous](#) [Next ▶](#)

n
t.
P
l
e
a
s
e
a
d
j
u
s
t
t
h
e
c
o
d
e
a
c
c
o
r
d
i
n
g
ly

[Reply](#)



JournalDev

< Previous Next >



include
the
filena
me
along
with
code
snippe
t. How
will
anyon
e
unders
tand
where
to add
the
code.
And
please
do this
for all
tutorial
s. All
tutorial
s on
this
websit
e have
the
same
proble
m

Reply



JournalDev

[◀ Previous](#) [Next ▶](#)

i
d
e
n
ti
f
y
t
h
e
fi
l
e
n
a
m
e
j
u
s
t
b
y
l
o
o
k
i
n
g
a
t
t
h



JournalDev

[◀ Previous](#) [Next ▶](#)

s
m
a
v
e
n
d
e
p
e
n
d
e
n
c
i
e
s,
it
c
a
n
o
n
l
y
b
e
p
o
m
.x
m



JournalDev

[◀ Previous](#) [Next ▶](#)

l
i
c
J
a
v
a
c
l
a
s
s,
t
h
e
n
y
o
u
k
n
o
w
t
h
a
t
t
h
e
f
i
l
e
n
a



JournalDev

[◀ Previous](#) [Next ▶](#)

s
s
n
a
m
e
it
s
e
lf
.M
o
r
e
o
v
e
r,
m
o
s
t
o
f
t
h
e
t
u
t
o
ri
a



JournalDev

[◀ Previous](#) [Next ▶](#)

w
n
l
o
a
d
a
b
l
e
z
i
p
f
l
e
s,
if
y
o
u
s
t
i
l
c
a
n'
t
g
e
t
it
t
h



JournalDev

< Previous Next >



t
e
l
s
e
w
il
Reply



Tainoor says:
**Bashir
Manvar**

UserRe
positor
y
findOn
e(Strin
g
UserID
)
metho
d is
not
availa
ble
with
spring
Reply

Gokul says:

February 18,
2019 at
11:37 pm

I
a



JournalDev

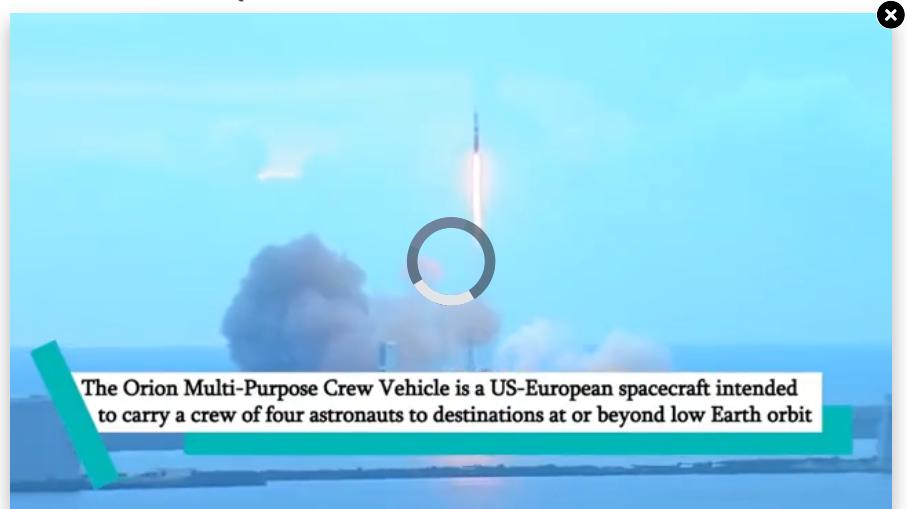
[◀ Previous](#) [Next ▶](#)

e
a
n
y
s
o
l
u
ti
o
n
Reply[?]

November jenkgoy says:

28,
2019
at
11:17
pm

H
o
w
d
i
d
y
o
u
s
o
l
v
e
d
t



JournalDev

[◀ Previous](#) [Next ▶](#)

e
s
a
m
e
i
s
s
u
e

Reply

Ravi says:
January
12,
2020
at
4:06
am

J
u
s
t
c
h
a
n
g
e
v
e
r
s
i
o
n
n



JournalDev

< Previous Next >

[Reply](#) [e](#)

August 12, 2018 at 10:45 pm [Omar Y](#) says:

Great
tutorial
! Very
Useful
!! [Reply](#)

July 10, 2018 at 9:54 pm [ddubey](#) says:

Great
article
Shubh
am. It's
quite
useful
for
people
who
want
to start
learnin
g
Mongo
DB
and
Spring
Boot
[Reply](#)



JournalDev

< Previous Next >



started
the
project
from
Spring
initializ
er and
went
throug
h your
tutorial
. the
findOn
e()
metho
d was
showin
g an
error. i
found
out
that
the
newer
version
s of
spring-
boot-
starter-
parent
release
s(what
I
downl



JournalDev

[◀ Previous](#) [Next ▶](#)

d. I am
going
with
the
1.5.8
version
for
now,
but
what
should
I do in
the
newer
version
so to
use the
findOn
method
d??
[Reply](#)

June 6, 2018 **sam** says:
at 2:46 pm Serg

M
a
y
b
e
i
a
m
t



JournalDev

[◀ Previous](#) [Next ▶](#)

v
e
r
s
i
o
n
s
y
o
u
n
e
e
d
t
o
u
s
e
“
u
s
e
r
R
e
p
o
s
it
o
r
y.



JournalDev

[◀ Previous](#) [Next ▶](#)

e
t(
);
"
r
a
t
h
e
r
t
h
a
n
"
u
s
e
r
R
e
p
o
s
it
o
r
y.
fi
n
d
O
n
e



JournalDev

[◀ Previous](#) [Next ▶](#)

o
u
a
r
e
r
i
g
h
t
S
a
m
.W
e
h
a
v
e
t
o
c
h
a
n
g
e
i
t
i
n
o
r



JournalDev

[◀ Previous](#) [Next ▶](#)10:00
am

I
n
n
e
w
e
r
v
e
r
s
i
o
n
w
e
c
a
n
u
s
e
l
i
k
e
t
h
i
s
o
p
t



JournalDev

[◀ Previous](#) [Next ▶](#)

e
p
o
s
i
t
o
r
y
.f
i
n
d
o
n
e
(



JournalDev

[◀ Previous](#) [Next ▶](#)

u
s
e
r
i
s
p
r
e
s
e
n
t
i
n
d
a
t
a
b
a
s
e
o
r
n
o
t
b
y
,
i
f
(



JournalDev

[◀ Previous](#) [Next ▶](#)

```
{  
U  
s  
e  
r  
u  
s  
e  
r  
D  
a  
t  
a  
=  
u  
s  
e  
r  
.g  
e  
t  
(  
)  
;  
/  
/  
U  
r  
l  
n  
i  
t  
i
```



JournalDev

< Previous Next >

r
e
:
)
...
. }
Reply }October
2,
2019
at
12:34
am

Devesh says:

t
h
a
n
k
y
o
u
Reply u

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name



Newsletter for You

Enter your email address here...



JournalDev

JournalDev is one of the most popular websites for Java, Python, Android, and related technical articles. Our tutorials are regularly updated, error-free, and complete. Every month millions of developers like you visit JournalDev to read our tutorials.

JournalDev was founded by Pankaj Kumar in 2010 to share his experience and learnings with the whole world. He loves Open source

Most Popular

Java / Java EE Tutorials

Core Java Tutorial

Python Tutorials

Java Interview Questions

Favorite Sites

AskPython

BloggerTipsAndTricks

GoLangDocs

LinuxForDevices

Mkyong



JournalDev

[◀ Previous](#) [Next ▶](#)

passion.

© 2020 · [Privacy Policy](#) · [Terms and Conditions](#) · [Contact Us](#) · [About](#) · Part of JournalDev IT Services Private Limited

