```
Yesterday's session : REST Controller Unit Testing
_____
-> We can perform Unit testing for rest controller using MockMvc.
MockMvcRequestBuilders reqBuilder =
                     MockMvcRequestBuilders.get("");
MvcResult result = mockMvc.perform(reqBuilder).andReturn();
-> We can validate rest api method results using response body conte
       int status = result.getResponse().getStatus();
What is Mocking?
-> It is the process of creating substitute/alternate object for the
package com.ashokit.service;
import org.springframework.stereotype.Service;
import com.ashokit.bindings.Book;
@Service
public class BookService {
       public boolean saveBook(Book book) {
             return true;
       }
package com.ashokit.rest;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springf
'-ind.annotation.PostMapping;
import org.springf
                            .nd.annotation.RequestBody;
```



```
Yesterday's session : REST Controller Unit Testing
_____
-> We can perform Unit testing for rest controller using MockMvc.
MockMvcRequestBuilders reqBuilder =
                     MockMvcRequestBuilders.get("");
MvcResult result = mockMvc.perform(reqBuilder).andReturn();
-> We can validate rest api method results using response body conte
       int status = result.getResponse().getStatus();
What is Mocking?
-> It is the process of creating substitute/alternate object for the
package com.ashokit.service;
import org.springframework.stereotype.Service;
import com.ashokit.bindings.Book;
@Service
public class BookService {
       public boolean saveBook(Book book) {
             return true;
       }
package com.ashokit.rest;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springf
'--'nd.annotation.PostMapping;
import org.springf
                            .nd.annotation.RequestBody;
```



```
Yesterday's session : REST Controller Unit Testing
_____
-> We can perform Unit testing for rest controller using MockMvc.
MockMvcRequestBuilders reqBuilder =
                     MockMvcRequestBuilders.get("");
MvcResult result = mockMvc.perform(reqBuilder).andReturn();
-> We can validate rest api method results using response body conte
       int status = result.getResponse().getStatus();
What is Mocking?
-> It is the process of creating substitute/alternate object for the
package com.ashokit.service;
import org.springframework.stereotype.Service;
import com.ashokit.bindings.Book;
@Service
public class BookService {
       public boolean saveBook(Book book) {
             return true;
       }
package com.ashokit.rest;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springf
'--'nd.annotation.PostMapping;
import org.springf
                            .nd.annotation.RequestBody;
```



```
Yesterday's session : REST Controller Unit Testing
_____
-> We can perform Unit testing for rest controller using MockMvc.
MockMvcRequestBuilders reqBuilder =
                     MockMvcRequestBuilders.get("");
MvcResult result = mockMvc.perform(reqBuilder).andReturn();
-> We can validate rest api method results using response body conte
       int status = result.getResponse().getStatus();
What is Mocking?
-> It is the process of creating substitute/alternate object for the
package com.ashokit.service;
import org.springframework.stereotype.Service;
import com.ashokit.bindings.Book;
@Service
public class BookService {
       public boolean saveBook(Book book) {
             return true;
       }
package com.ashokit.rest;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springf
'--'nd.annotation.PostMapping;
import org.springf
                            .nd.annotation.RequestBody;
```



```
Yesterday's session : REST Controller Unit Testing
_____
-> We can perform Unit testing for rest controller using MockMvc.
MockMvcRequestBuilders reqBuilder =
                     MockMvcRequestBuilders.get("");
MvcResult result = mockMvc.perform(reqBuilder).andReturn();
-> We can validate rest api method results using response body conte
       int status = result.getResponse().getStatus();
What is Mocking?
-> It is the process of creating substitute/alternate object for the
package com.ashokit.service;
import org.springframework.stereotype.Service;
import com.ashokit.bindings.Book;
@Service
public class BookService {
       public boolean saveBook(Book book) {
             return true;
       }
package com.ashokit.rest;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springf
'--'nd.annotation.PostMapping;
import org.springf
                            .nd.annotation.RequestBody;
```



```
Yesterday's session : REST Controller Unit Testing
_____
-> We can perform Unit testing for rest controller using MockMvc.
MockMvcRequestBuilders reqBuilder =
                     MockMvcRequestBuilders.get("");
MvcResult result = mockMvc.perform(reqBuilder).andReturn();
-> We can validate rest api method results using response body conte
       int status = result.getResponse().getStatus();
What is Mocking?
-> It is the process of creating substitute/alternate object for the
package com.ashokit.service;
import org.springframework.stereotype.Service;
import com.ashokit.bindings.Book;
@Service
public class BookService {
       public boolean saveBook(Book book) {
             return true;
       }
package com.ashokit.rest;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springf
'--'nd.annotation.PostMapping;
import org.springf
                            .nd.annotation.RequestBody;
```



```
Yesterday's session : REST Controller Unit Testing
_____
-> We can perform Unit testing for rest controller using MockMvc.
MockMvcRequestBuilders reqBuilder =
                     MockMvcRequestBuilders.get("");
MvcResult result = mockMvc.perform(reqBuilder).andReturn();
-> We can validate rest api method results using response body conte
       int status = result.getResponse().getStatus();
What is Mocking?
-> It is the process of creating substitute/alternate object for the
package com.ashokit.service;
import org.springframework.stereotype.Service;
import com.ashokit.bindings.Book;
@Service
public class BookService {
       public boolean saveBook(Book book) {
             return true;
       }
package com.ashokit.rest;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springf
'--'nd.annotation.PostMapping;
import org.springf
                            .nd.annotation.RequestBody;
```



```
Yesterday's session : REST Controller Unit Testing
_____
-> We can perform Unit testing for rest controller using MockMvc.
MockMvcRequestBuilders reqBuilder =
                     MockMvcRequestBuilders.get("");
MvcResult result = mockMvc.perform(reqBuilder).andReturn();
-> We can validate rest api method results using response body conte
       int status = result.getResponse().getStatus();
What is Mocking?
-> It is the process of creating substitute/alternate object for the
package com.ashokit.service;
import org.springframework.stereotype.Service;
import com.ashokit.bindings.Book;
@Service
public class BookService {
       public boolean saveBook(Book book) {
             return true;
       }
package com.ashokit.rest;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springf
'-ind.annotation.PostMapping;
import org.springf
                            .nd.annotation.RequestBody;
```



```
Yesterday's session : REST Controller Unit Testing
_____
-> We can perform Unit testing for rest controller using MockMvc.
MockMvcRequestBuilders reqBuilder =
                     MockMvcRequestBuilders.get("");
MvcResult result = mockMvc.perform(reqBuilder).andReturn();
-> We can validate rest api method results using response body conte
       int status = result.getResponse().getStatus();
What is Mocking?
-> It is the process of creating substitute/alternate object for the
package com.ashokit.service;
import org.springframework.stereotype.Service;
import com.ashokit.bindings.Book;
@Service
public class BookService {
       public boolean saveBook(Book book) {
             return true;
       }
package com.ashokit.rest;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.http.ResponseEntity;
import org.springf
'-ind.annotation.PostMapping;
import org.springf
                            .nd.annotation.RequestBody;
```



```
Yesterday's session : REST Controller Unit Testing
           -> We can perform Unit testing for rest controller using MockMvc.
           MockMvcRequestBuilders reqBuilder =
                                  MockMvcRequestBuilders.get("");
           MvcResult result = mockMvc.perform(reqBuilder).andReturn();
            -> We can validate rest api method results using response body conte
                   int status = result.getResponse().getStatus();
           What is Mocking?
            -> It is the process of creating substitute/alternate object for the
           package com.ashokit.service;
            import org.springframework.stereotype.Service;
           import com.ashokit.bindings.Book;
      Asho
           @Service
      Dec 4
           public class BookService {
04-Dec-2020 C
                   public boolean saveBook(Book book) {

    ○4-Dec

                          return true;
                   }
            package com.ashokit.rest;
           import org.springframework.beans.factory.annotation.Autowired;
 12-JRTP@
            import org.springframework.http.HttpStatus;
            import org.springframework.http.ResponseEntity;
            import org.springf
'-ind.annotation.PostMapping;
            import org.springf
                                         .nd.annotation.RequestBody;
```

```
Yesterday's session : REST Controller Unit Testing
             -> We can perform Unit testing for rest controller using MockMvc.
      Asho
      Dec 2
            MockMvcRequestBuilders reqBuilder =
02-Dec-2020 c
                                     MockMvcRequestBuilders.get("");
  🖣 02-Dec
             MvcResult result = mockMvc.perform(reqBuilder).andReturn();
             -> We can validate rest api method results using response body conte
                     int status = result.getResponse().getStatus();
      Asho
            What is Mocking?
      Dec 1
01-Dec-2020 C
             -> It is the process of creating substitute/alternate object for the
    01-Dec
             package com.ashokit.service;
             import org.springframework.stereotype.Service;
             import com.ashokit.bindings.Book;
             @Service
      Asho
            public class BookService {
      Nov 3
                    public boolean saveBook(Book book) {
30-Nov-2020 C
                            return true;
                     }
    30-Nov
            package com.ashokit.rest;
             import org.springframework.beans.factory.annotation.Autowired;
 12-JRTP@
             import org.springframework.http.HttpStatus;
             import org.springframework.http.ResponseEntity;
             import org.springf
'-ind.annotation.PostMapping;
             import org.springf
                                             .nd.annotation.RequestBody;
             import org springf..... .... ... ... and annotation RestController:
```

```
Yesterday's session : REST Controller Unit Testing
            -> We can perform Unit testing for rest controller using MockMvc.
From 23rd onv
            MockMvcRequestBuilders reqBuilder =
                                   MockMvcRequestBuilders.get("");
1 class comm
            MvcResult result = mockMvc.perform(reqBuilder).andReturn();
     Bijend
            -> We can validate rest api method results using response body conte
     Sir, ple
                    int status = result.getResponse().getStatus();
            What is Mocking?
            -> It is the process of creating substitute/alternate object for the
      Asho
28-Nov-2020 C
            package com.ashokit.service;
            import org.springframework.stereotype.Service;
            import com.ashokit.bindings.Book;
            @Service
            public class BookService {
      Kund
      Nov 2
                    public boolean saveBook(Book book) {
Sir,
                           return true;
please upload
                    }
            package com.ashokit.rest;
            import org.springframework.beans.factory.annotation.Autowired;
 12-JRTP@
            import org.springframework.http.HttpStatus;
            import org.springframework.http.ResponseEntity;
            import org.springf
'-ind.annotation.PostMapping;
            import org.springf
                                          .nd.annotation.RequestBody;
```

```
3 class comments
     Ganga
             Yesterday's session : REST Controller Unit Testing
     No att
             -> We can perform Unit testing for rest controller using MockMvc.
             MockMvcRequestBuilders reqBuilder =
                                     MockMvcRequestBuilders.get("");
      Asho
      Nov 2
             MvcResult result = mockMvc.perform(reqBuilder).andReturn();
22-Nov-2020 C
             -> We can validate rest api method results using response body conte
  22-Nov
                     int status = result.getResponse().getStatus();
4 class comm
             What is Mocking?
     siri 20
     Sir, Pl
             -> It is the process of creating substitute/alternate object for the
             package com.ashokit.service;
             import org.springframework.stereotype.Service;
             import com.ashokit.bindings.Book;
             @Service
             public class BookService {
                     public boolean saveBook(Book book) {
                             return true;
                     }
             package com.ashokit.rest;
             import org.springframework.beans.factory.annotation.Autowired;
             import org.springframework.http.HttpStatus;
             import org.springframework.http.ResponseEntity;
             import org.springf
'--'nd.annotation.PostMapping;
             import org.springf
                                             .nd.annotation.RequestBody;
```