

Title: Spring boot application fails after start - An attempt was made to call a method that does not exist. The attempt was made from the following location

Post Body:

I'm new in Spring Boot, I'm trying to build a simple application with REST Api, and I'm getting this error immediately when starting the app:

Error starting ApplicationContext. To display the conditions report re-run your application with 'debug' enabled. 2021-01-24 2

I did not finish yet with the development of the app, but I can't go on because of this errors.

There are my classes so far:

pom.xml

```
<?xml version="1.0" encoding="UTF-8"?> <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <groupId>com.example</groupId>
    <artifactId>micro1</artifactId>
    <version>1.0.0</version>
    <packaging>jar</packaging>
    <name>micro1</name>
    <description>A simple Spring Boot application</description>
    <url>http://example.com</url>
    <licenses>
        <license>
            <name>MIT License</name>
            <url>http://www.opensource.org/licenses/mit-license.php</url>
        </license>
    </licenses>
    <developers>
        <developer>
            <name>John Doe</name>
            <email>john.doe@example.com</email>
            <organization>Example Corp.</organization>
        </developer>
    </developers>
    <dependencies>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter</artifactId>
            <version>2.2.1</version>
        </dependency>
        <dependency>
            <groupId>org.springframework.boot</groupId>
            <artifactId>spring-boot-starter-web</artifactId>
            <version>2.2.1</version>
        </dependency>
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-jdbc</artifactId>
            <version>5.2.1</version>
        </dependency>
        <dependency>
            <groupId>com.h2database</groupId>
            <artifactId>h2</artifactId>
            <version>1.4.200</version>
        </dependency>
    </dependencies>
    <build>
        <plugins>
            <plugin>
                <groupId>org.springframework.boot</groupId>
                <artifactId>spring-boot-maven-plugin</artifactId>
            </plugin>
        </plugins>
    </build>
</project>
```

controller:

```
package com.example.micro1.controllers; import com.example.micro1.entities.User; import com.example.micro1.services.Service;

import org.springframework.web.bind.annotation.RequestMapping; import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping("/api")
public class UserController {

    private Service service;

    public UserController(Service service) {
        this.service = service;
    }

    @GetMapping("/users")
    public List<User> getUsers() {
        return service.getAllUsers();
    }

    @PostMapping("/users")
    public User createUser(@RequestBody User user) {
        return service.createUser(user);
    }

    @PutMapping("/users/{id}")
    public User updateUser(@PathVariable Long id, @RequestBody User user) {
        return service.updateUser(id, user);
    }

    @DeleteMapping("/users/{id}")
    public boolean deleteUser(@PathVariable Long id) {
        return service.deleteUser(id);
    }
}
```

service:

```
package com.example.micro1.services; import com.example.micro1.entities.User; import com.example.micro1.repositories.Repository;

import org.springframework.stereotype.Service;

@Service
public class Service {

    private Repository repository;

    public Service(Repository repository) {
        this.repository = repository;
    }

    public List<User> getAllUsers() {
        return repository.findAll();
    }

    public User createUser(User user) {
        return repository.save(user);
    }

    public User updateUser(Long id, User user) {
        User existingUser = repository.findById(id).orElseThrow(() -> new RuntimeException("User not found"));
        existingUser.setName(user.getName());
        existingUser.setEmail(user.getEmail());
        return repository.save(existingUser);
    }

    public boolean deleteUser(Long id) {
        repository.deleteById(id);
        return true;
    }
}
```

repository:

```
package com.example.micro1.repositories; import com.example.micro1.entities.User;

import org.springframework.data.jpa.repository.JpaRepository;

public interface Repository extends JpaRepository<User, Long> {
}
```

One entity for now:

```
package com.example.micro1.entities; import javax.persistence.Entity; import javax.persistence.GeneratedValue; import javax.persistence.GenerationType;

import org.hibernate.annotations.GenericGenerator;

@Entity
@GenericGenerator(name = "uuid", strategy = GenerationType.UUID)
public class User {

    @GeneratedValue(generator = "uuid")
    @GenericGenerator(name = "uuid", strategy = GenerationType.UUID)
    private Long id;

    private String name;

    private String email;

    // getters and setters
}

```

application.properties:

```
#h2
spring.datasource.url=jdbc:h2:mem:testdb
spring.datasource.driverClassName=org.h2.Driver
spring.datasource.username=sa
spring.datasource.password=
spring.h2.console.enabled=true
```

The application class:

```
package com.example.micro1; import org.springframework.boot.SpringApplication; import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class Micro1Application {

    public static void main(String[] args) {
        SpringApplication.run(Micro1Application.class, args);
    }
}
```

Please tell me if there is something else that I should write here... Thanks!

Accepted Answer:

Your starter spring-boot-starter-web contains inside itself these dependencies

```
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-web</artifactId>
    <version>5.3.3</version>
    <scope>compile</scope>
</dependency>
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-webmvc</artifactId>
    <version>5.3.3</version>
    <scope>compile</scope>
</dependency>
```

besides, you add in your pom.xml that:

```
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-webmvc</artifactId>
    <version>5.3.3</version>
    <scope>compile</scope>
</dependency>
```

so, when you'll start your application you have in classpath ambiguity. I think it's cause of your problem. Spring Boot starters save you the trouble of specifying explicitly declare dependency. Try to remove spring-webmvc dependency.

Highest Rated Answer:

I had this same issue when I run a Spring Boot Application of Web. But I got following error messages:

```
***** APPLICATION FAILED TO START *****
Description: An attempt was made to call the method public void org.springframework.boot.autoconfigure.jdbc.DataSourceConfiguration.setDriverClassName(String) which does not exist.
```

This may be spring-boot-autoconfigure's exception, so I have checked the dependency of it:

```
org.springframework.boot:spring-boot-web:2.2.1.RELEASE-->
org.springframework.boot:spring-boot-starter:2.2.1.RELEASE -->
org.springframework.boot:spring-boot-autoconfigure:2.2.1.RELEASE(omitted for conflict with 2.1.6.RELEASE).
```

Obviously, the autoconfigure was failing because of a version conflict. To solve this problem, I have added the following dependency:

```
<dependency>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-autoconfigure</artifactId>
    <version>2.2.1</version>
    <scope>compile</scope>
</dependency>
```

I've confirmed this dependency had been added in the module and I've rerun the application.