

Come implementare l'upload di file multipli con Java

L'upload di file multipli è una funzionalità comune in molte applicazioni web moderne. In questo articolo, esploreremo come implementare questa funzionalità utilizzando Java, sfruttando il framework Spring Boot.

Requisiti

Prima di iniziare, assicurati di avere i seguenti requisiti:

- Java 8 o superiore
- Spring Boot
- Maven o Gradle

Configurazione del Progetto

Inizia creando un nuovo progetto Spring Boot. Aggiungi la seguente dipendenza al file `pom.xml`:

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-web</artifactId>
</dependency>
```

Creazione del Controller

Crea un controller per gestire l'upload dei file:

```

@RestController
@RequestMapping("/upload")
public class FileUploadController {

    @PostMapping("/multiple")
    public ResponseEntity<String>
uploadMultipleFiles(@RequestParam("files")
MultipartFile[] files) {
        StringBuilder message = new StringBuilder();

        for (MultipartFile file : files) {
            // Logica per gestire il file
            message.append("File caricato:
").append(file.getOriginalFilename()).append("\n");
        }

        return ResponseEntity.ok(message.toString());
    }
}

```

Aggiunta di un Form HTML

Per testare l'endpoint, aggiungi un semplice form HTML:

```

<form method="POST" action="/upload/multiple"
enctype="multipart/form-data">
    <input type="file" name="files" multiple />
    <button type="submit">Upload</button>
</form>

```

Gestione dei File

Puoi salvare i file su disco o elaborarli in altro modo. Ecco un esempio per salvarli:

```
for (MultipartFile file : files) {  
    Path path = Paths.get("uploads/" +  
file.getOriginalFilename());  
    Files.write(path, file.getBytes());  
}
```

Conclusione

Con pochi passaggi, puoi implementare l'upload di file multipli in Java. Spring Boot semplifica notevolmente il processo, rendendo il codice leggibile e manutenibile.