

# Download



Download from  
**Dreamstime.com**  
The illustrations come through in the processing purposes only.



2468711  
Milan Surkala | Dreamstime.com

---

mevgdev. February 15, 2020 at 2:00 am. . /12077699-youtube-video-downloader-pro-v14-final-4-7-filehippo A: The proper regex for this is `var test = "/3037778-sulejman-bulgari-vratimo-se-gospodaru-pdf-download14-15-ludarma /2992986-exclusive-youtube-video-downloader-pro-v15-13-4-9-final-utorrent /3412985-sulejman-bulgari-vratimo-se-gospodaru-pdf-download14-15-meygdev /12077699-youtube-video-downloader-pro-v14-final-4-7-filehippo"; var regex = new Regex(@"\V(?:+)\V"); var match = regex.Match(test); if (match.Success) { var result = match.Groups["file"].Value; }` This is assuming that you want to keep the regular expression code as short as possible. In fact, in this case it's quite short, so you can just keep it as a variable. The problem with your regex is that it contains a lot of backslashes that would not be needed with a proper regular expression. `\` is a meta character, and the string is treated literally as you can see in this regex: `Regex regex = new Regex(@"\V(?:+)\V");` Backslash will be treated literally and not as an escape character. To fix this problem, you need to escape the backslash. Therefore, a proper regex for the same code would be: `var regex = new Regex(@"\V(?:.*)\V");` And this will solve your problem because you don't have to escape the backslashes. Hope this helped :) A: It is because `V` is interpreted as `/`. You should escape it by using regex like `var regex = new Regex(@"\V(?:+)\V");` Q: Spring Data JDBC Database 'Connections' not found I 2d92ce491b