

# 正则表达式

关键词: Regular Expression, RegExp, 正则表达式

## 工具

<http://www.myregexp.com>

<https://regex101.com/>

## 实例

匹配Id, 但orderId和productId不算, 必须包含其他的Id (包含orderId或productId也可以, 但不能只有orderId或productId)

The screenshot shows a browser window with the URL [www.myregexp.com](http://www.myregexp.com). The page title is "Online regex tester". The main content area contains a text input field with the following regular expression: `(.*Id.*orderId.*)((.*orderId.*Id.*)((.*Id.*productId.*)((.*productId.*Id.*)((?!.*productId)^.*(?!.*orderId)^.*Id.*`. There are red annotations: a '1' pointing to the `(?!.*productId)^` part and a '2' pointing to the `(?!.*orderId)^` part. Below the input field, there are checkboxes for "Case insensitive (i)", "Global (g)", and "Multi-line (m)". The "Text to test" field contains the URL: `https://biz.alibaba.com/contract/sorSell.htm?123&buyerAccountId=710766709&orderId=xx&buyerAccountId=71076670&orderId=000&buyerAccountId=710766709`. Below the text field, it says "1 matches found". The "Capture groups" section shows the following results: #0: `https://biz.alibaba.com/contract/sorSell.htm?123&buyerAccountId=710766709&orderId=xx&buyerAccountId=71076670&orderId=000&buyerAccountId=710766709`, #1: `null`, #2: `https://biz.alibaba.com/contract/sorSell.htm?123&buyerAccountId=710766709&orderId=xx&buyerAccountId=71076670&orderId=000&buyerAccountId=710766709`, #3: `null`, #4: `null`.

参考: <http://stackoverflow.com/questions/1395177/regex-to-exclude-a-specific-string-constant>

I was helping a friend recently to do something very similar. However, he didn't want to match the string if it contained a string anywhere inside of it. So I wrote a slightly modified version of your expression `(?!.*ABC)^.*$` and this works like a charm. – Steve Wortham Apr 22 '10 at 15:41

另一种解决方案

使用 `(?!X)`, negative lookahead

<https://docs.oracle.com/javase/7/docs/api/java/util/regex/Pattern.html>

<code>(?=X)</code>	X, via zero-width positive lookahead
<code>(?!X)</code>	X, via zero-width negative lookahead
<code>(?&lt;=X)</code>	X, via zero-width positive lookbehind
<code>(?&lt;!X)</code>	X, via zero-width <b>negative lookbehind</b>

<http://www.regexg.com/regex-lookarounds.html>

Lookaround	Name	What it Does
(?=foo)	Lookahead	Asserts that what immediately follows the current position in the string is <i>foo</i>
(?<=foo)	Lookbehind	Asserts that what immediately precedes the current position in the string is <i>foo</i>
(?!foo)	Negative Lookahead	Asserts that what immediately follows the current position in the string is not <i>foo</i>
(?<!foo)	Negative Lookbehind	Asserts that what immediately precedes the current position in the string is not <i>foo</i>

REGULAR EXPRESSION: `(?<order|product|page)Id=`

TEST STRING: `https://biz.alibaba.com/contract/sorSell.htm?123&buyerAccountIsd=710766709&orderId=xx&productId=adff&buyerAccountId=aa|`

另一个例子:

Find in Path: `(?!import)\.FundCollector[:;]+`

84 matches in 12 files

- `@FundCollector(domain = "LOAN");` CreateLoanAction.java 103
- `FundCollectorLog.bizId(String.valueOf(createLoanRequest.getOrderid()));` CreateLoanAction.java 111
- `FundCollectorLog.bizType(createLoanRequest.getNyseBizType());` CreateLoanAction.java 112
- `FundCollectorLog.amount(createLoanRequest.getLoanAmount().getAmount());` CreateLoanAction.java 114
- `FundCollectorLog.amountCurrency(createLoanRequest.getLoanAmount().getCurrencyCode());` CreateLoanAction.java 115
- `FundCollectorLog.sourceEventBizId(createLoanRequest.getChannelInstrumentId());` CreateLoanAction.java 117
- `FundCollectorLog.sourceEventBizType("CHANNEL_INSTRUMENT");` CreateLoanAction.java 118
- `FundCollectorLog.parentEventBizId(String.valueOf(createLoanRequest.getForexdid()));` CreateLoanAction.java 119
- `FundCollectorLog.parentEventBizType("INCOME");` CreateLoanAction.java 120
- `@FundCollector(bizId = "${fundIncome.bizId}");` FundIncomeAdaptor.java 61
- `FundCollectorLog.amount(new BigDecimal(fundIncome.getAmount().getNumberStripped().toString()));` FundIncomeAdaptor.java 72

参考: <https://stackoverflow.com/questions/406230/regular-expression-to-match-a-line-that-doesnt-contain-a-word>