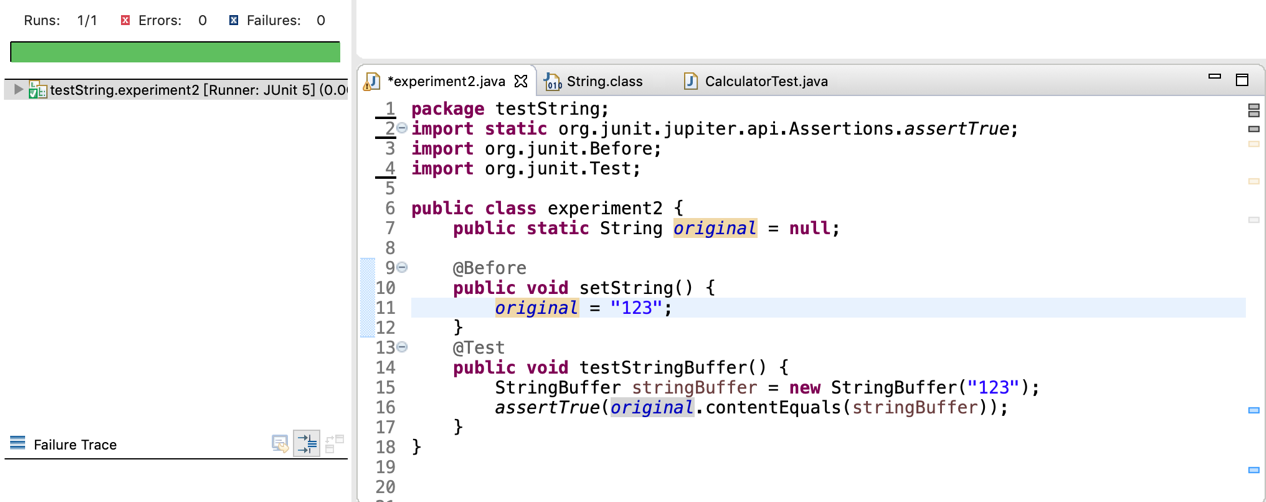
1. 测试public boolean contentEquals(CharSequence cs)
2. 输入相等的StringBuffer

分别构造一个”123”的字符串和StringBuffer对象

使用@Before 延迟加载构造的字符串

assertTrue测试后成功

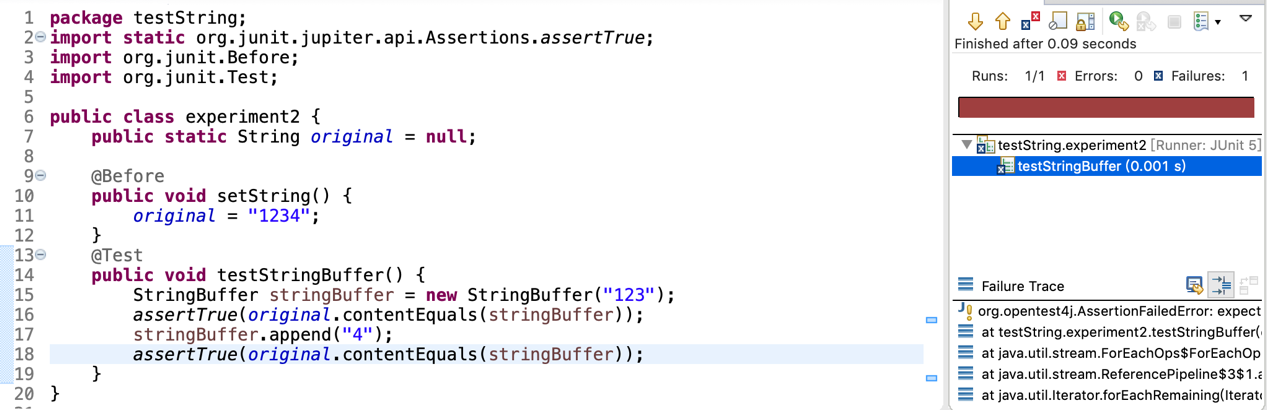


1. 输入一个不相等的StringBuffer

分别构造一个”123”的字符串和”1234“的StringBuffer对象

使用@Before 延迟加载构造的字符串

assertTrue测试后返回false

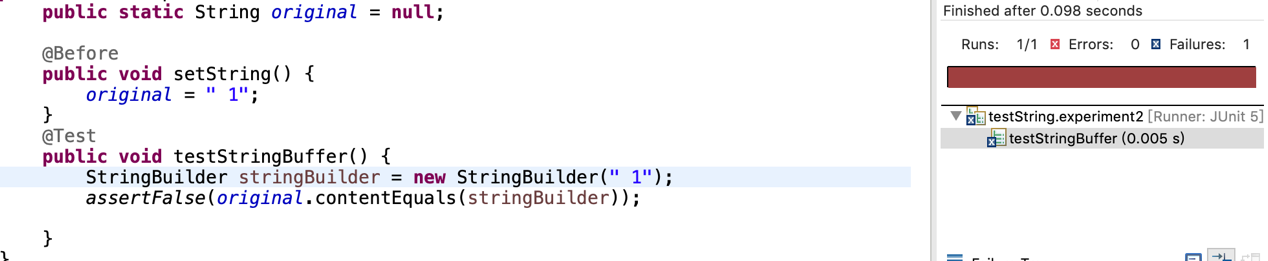


1. 输入一个相等的StringBuilder

分别构造一个” 1”的字符串和” 1“的StringBuffer对象

使用@Before 延迟加载构造的字符串

assertFalse测试后返回false

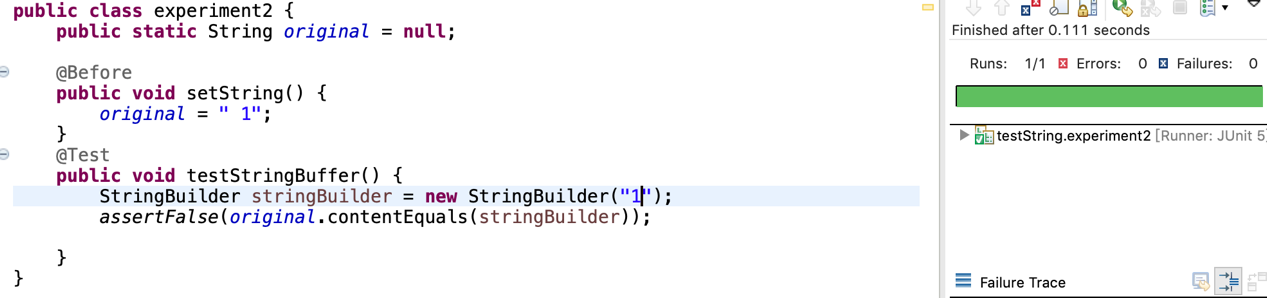


1. 输入一个不相等的StringBuilder

分别构造一个”1”的字符串和” 1“的StringBuffer对象

使用@Before 延迟加载构造的字符串

assertFalse测试后返回true

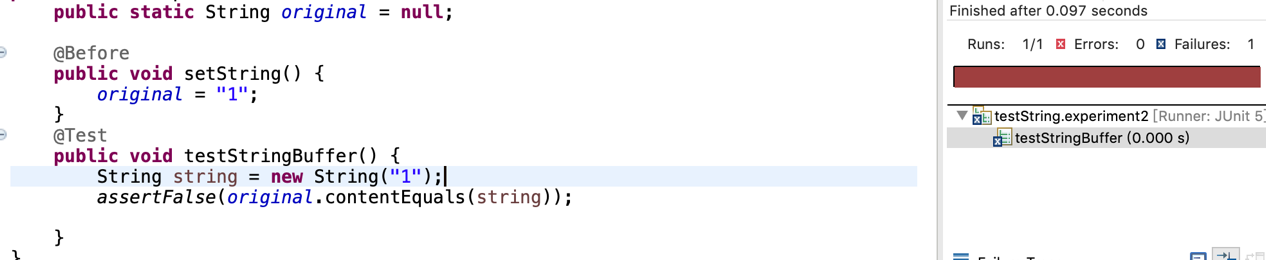


1. 输入一个相等的字符串

分别构造一个”1”的字符串和”1“的String对象

使用@Before 延迟加载构造的字符串

assertFalse测试后返回false

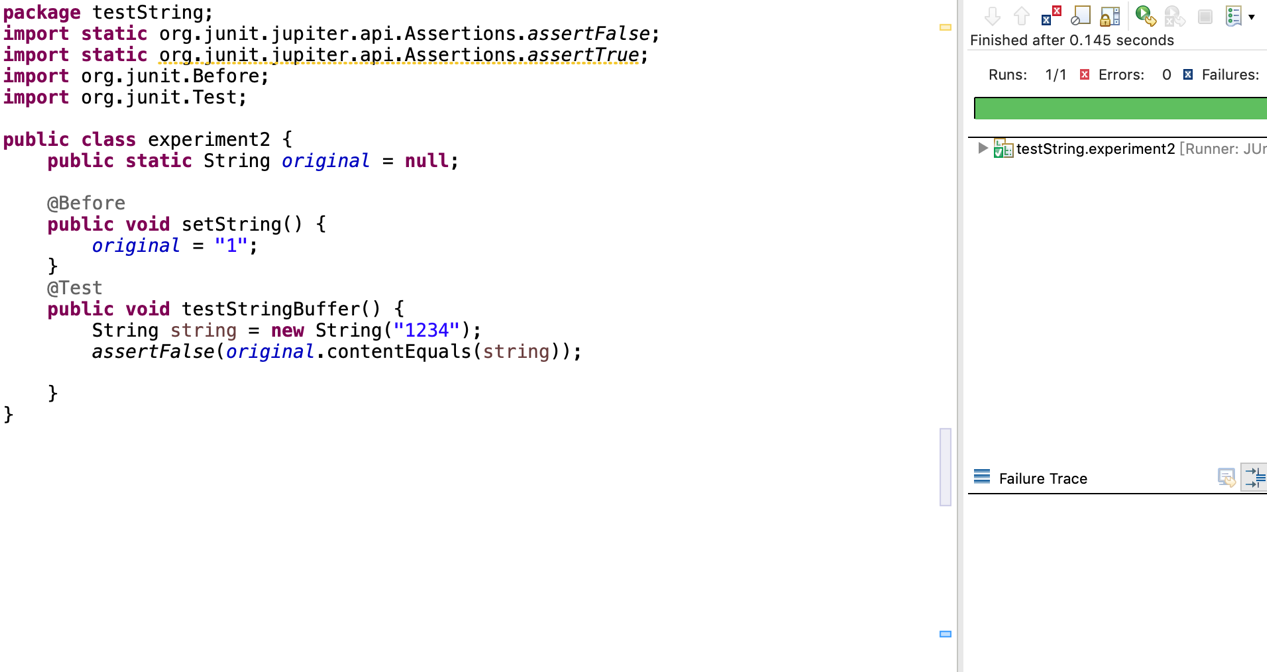


1. 输入一个不相等的字符串

分别构造一个”1”的字符串和”1“的String对象

使用@Before 延迟加载构造的字符串

assertFalse测试后返回true

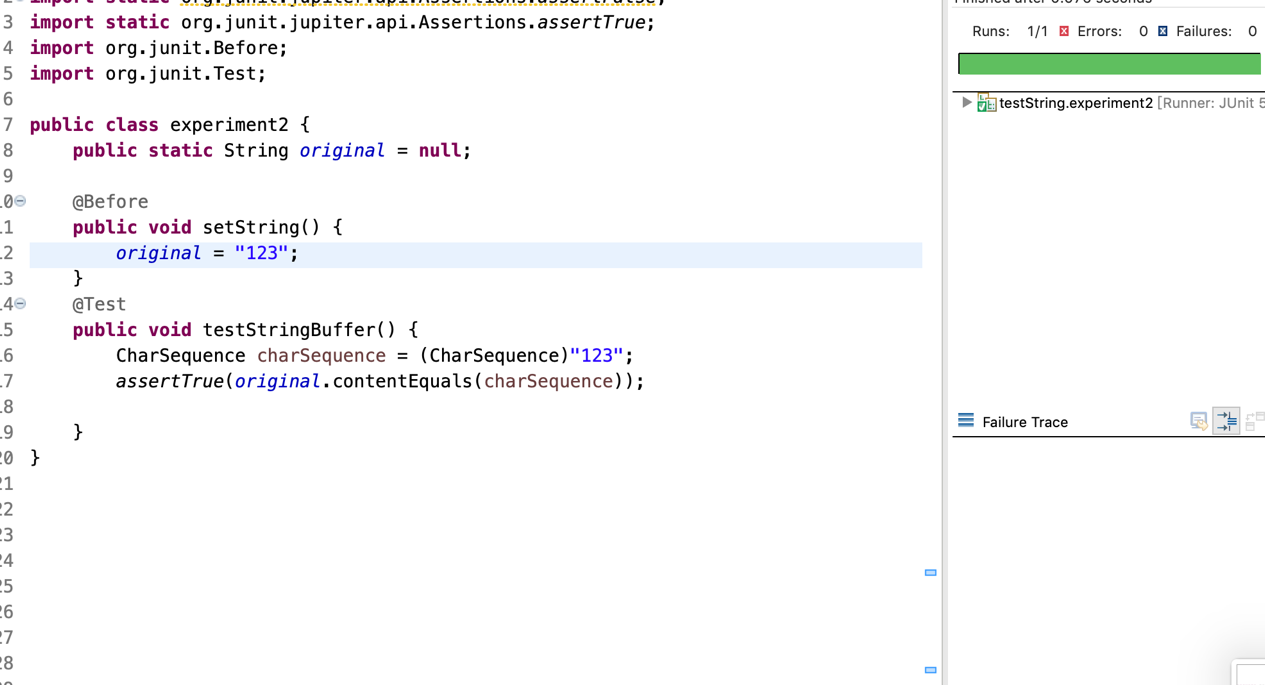


1. 输入一个相等的charSequence

分别构造一个”1”的字符串和”1“的Charsequence对象

使用@Before 延迟加载构造的字符串

assertTrue测试后返回true

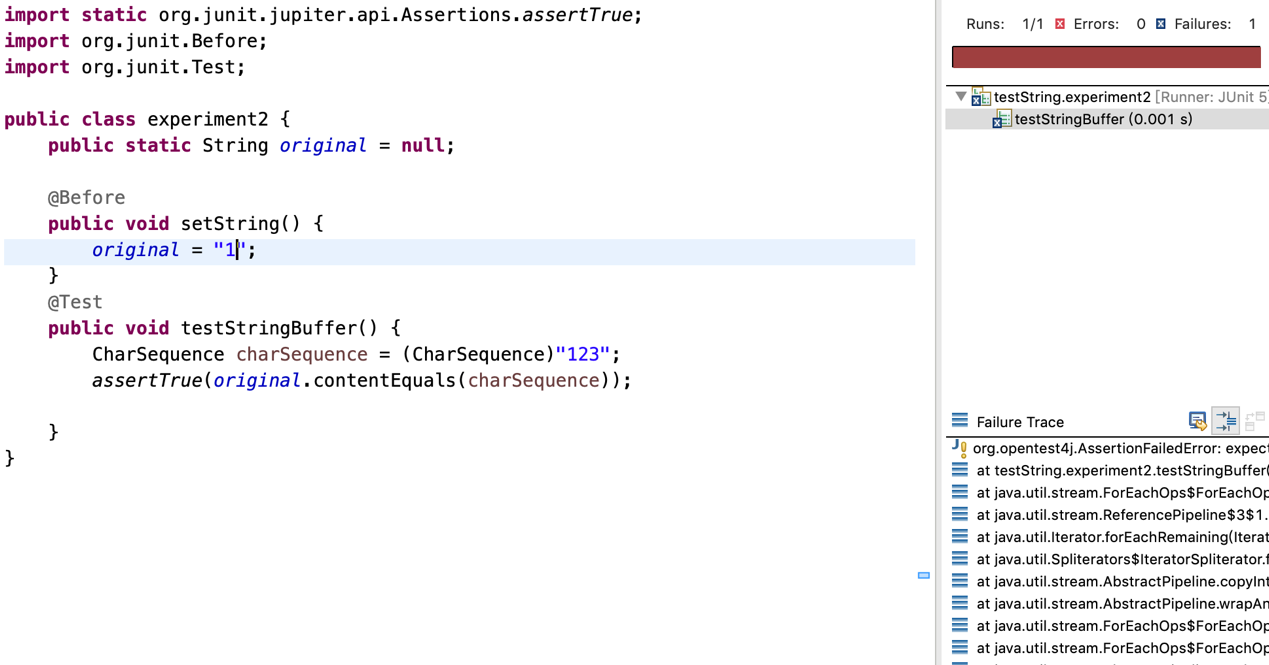


1. 输入一个不相等的charSequence

分别构造一个”123”的字符串和”1“的Charsequence对象

使用@Before 延迟加载构造的字符串

assertTrue测试后返回false

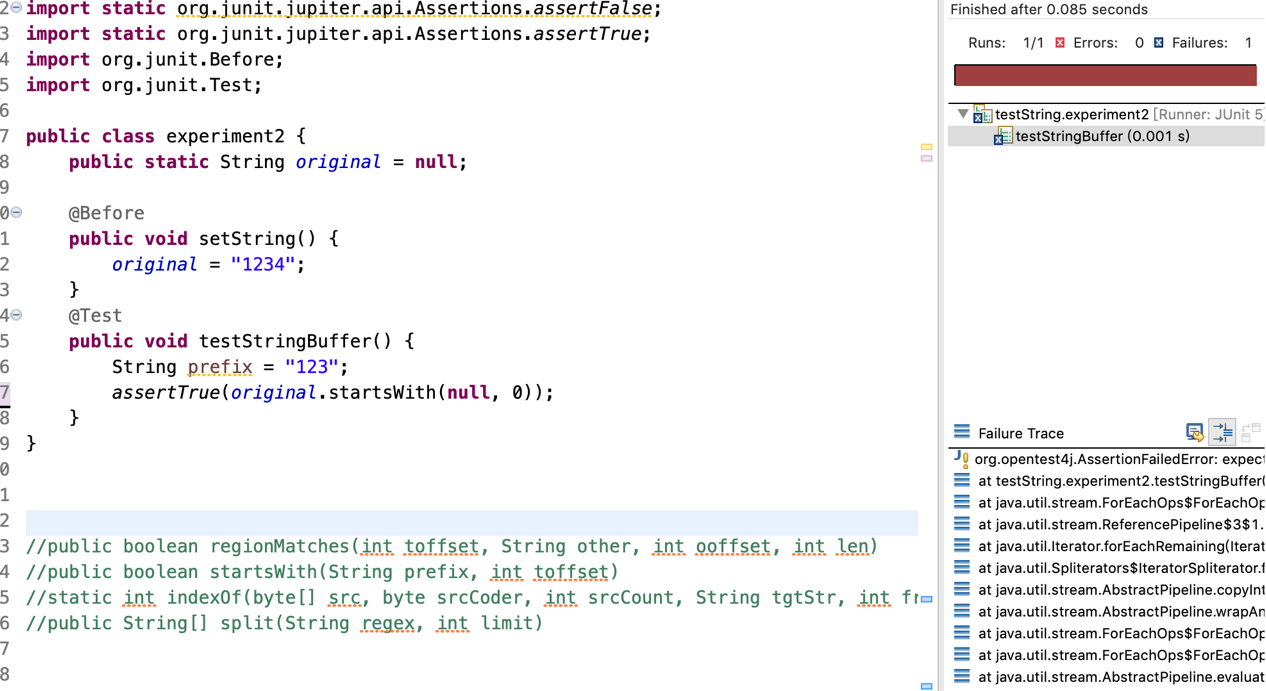


1. 测试 public Boolean startsWith(String prefix, int toffset)
2. 输入参数prefix为null

构造一个”123”的original字符串和一个”123”的prefix字符串，toffset为0

使用@Before 延迟加载构造的字符串

assertTrue测试后返回false



1. 输入参数toffset为负数

构造一个”123”的original字符串和一个”123”的prefix字符串，toffset为-1

使用@Before 延迟加载构造的字符串

assertTrue测试后返回false



构造一个”123”的original字符串和一个”123”的prefix字符串

使用@Before 延迟加载构造的字符串

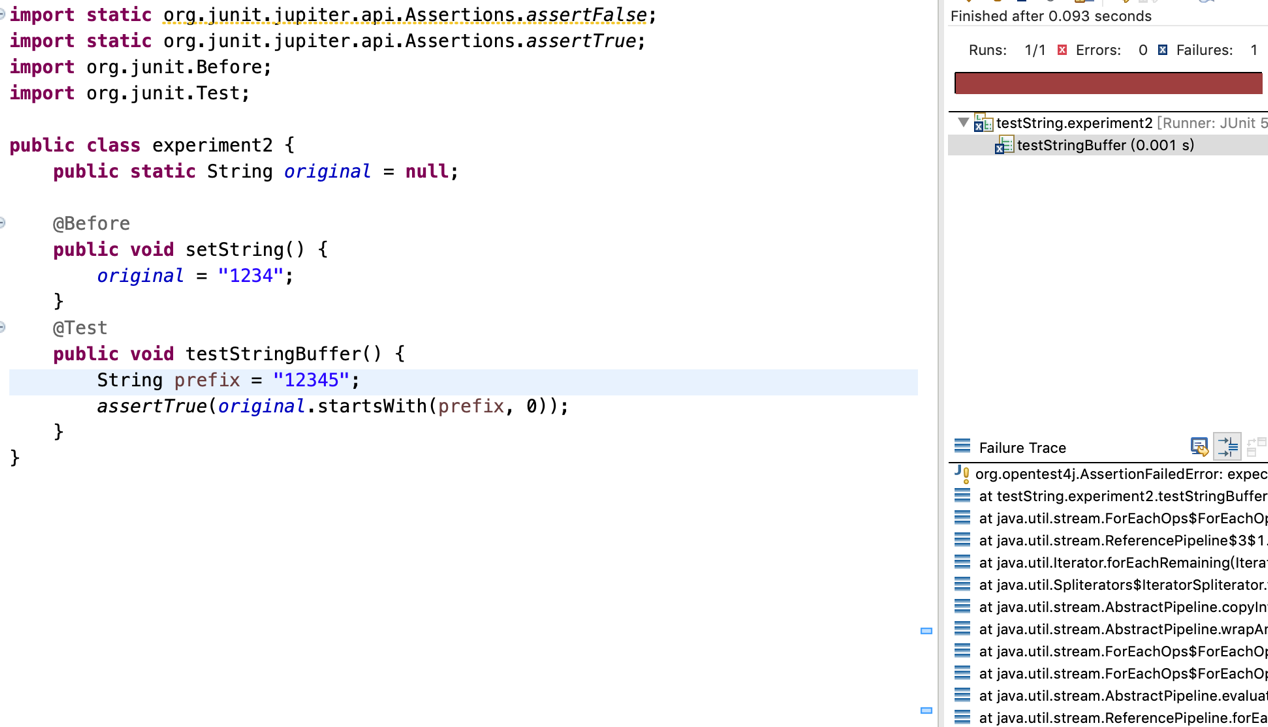
assertTrue测试后返回false

1. prefix的长度比original大

构造一个”1234”的original字符串和一个”12345”的prefix字符串，toffset为0

使用@Before 延迟加载构造的字符串

assertTrue测试后返回false

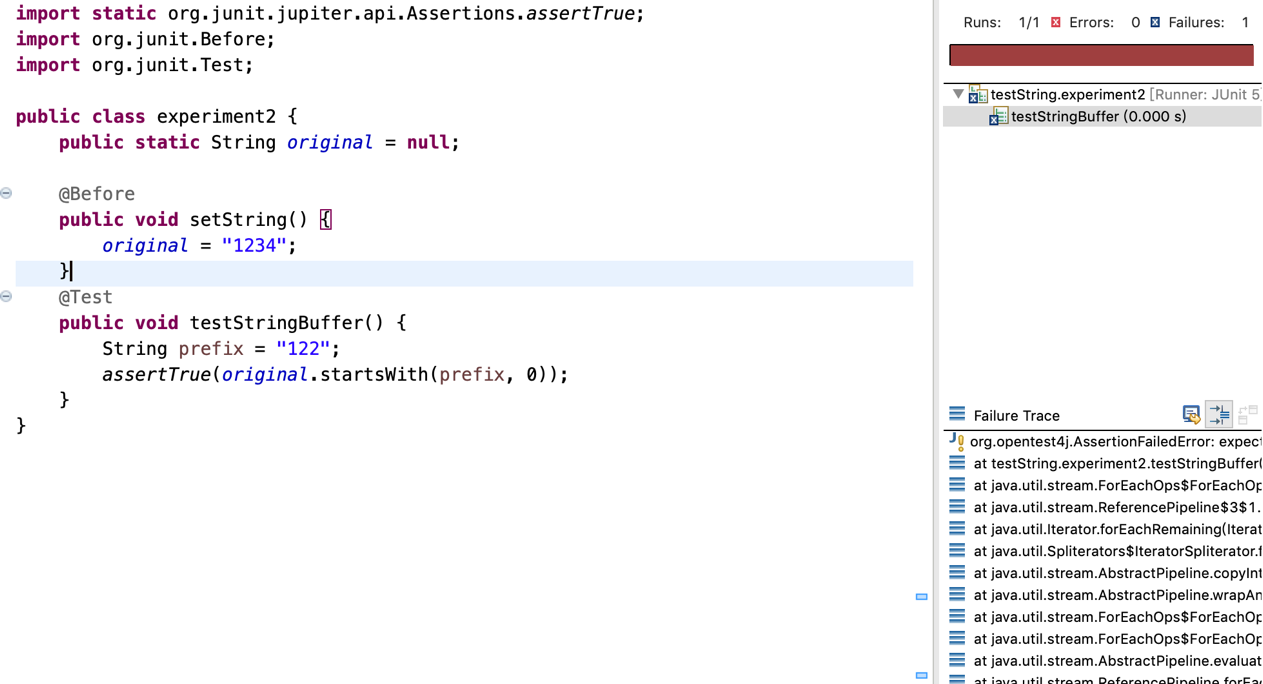


1. original上不存在prefix子串

构造一个”1234”的original字符串和一个”122”的prefix字符串，toffset为0

使用@Before 延迟加载构造的字符串

assertTrue测试后返回false

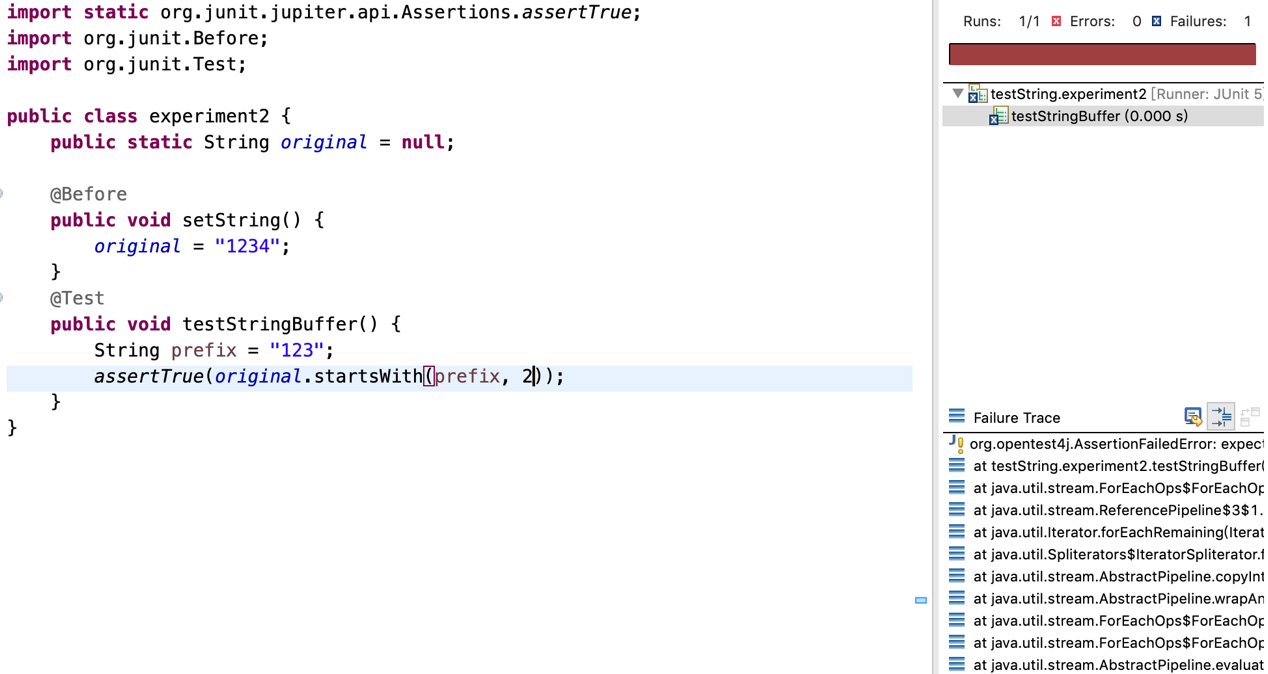


1. original存在prefix子串，但是toffset过大

构造一个”1234”的original字符串和一个”123”的prefix字符串，toffset为2

使用@Before 延迟加载构造的字符串

assertTrue测试后返回false

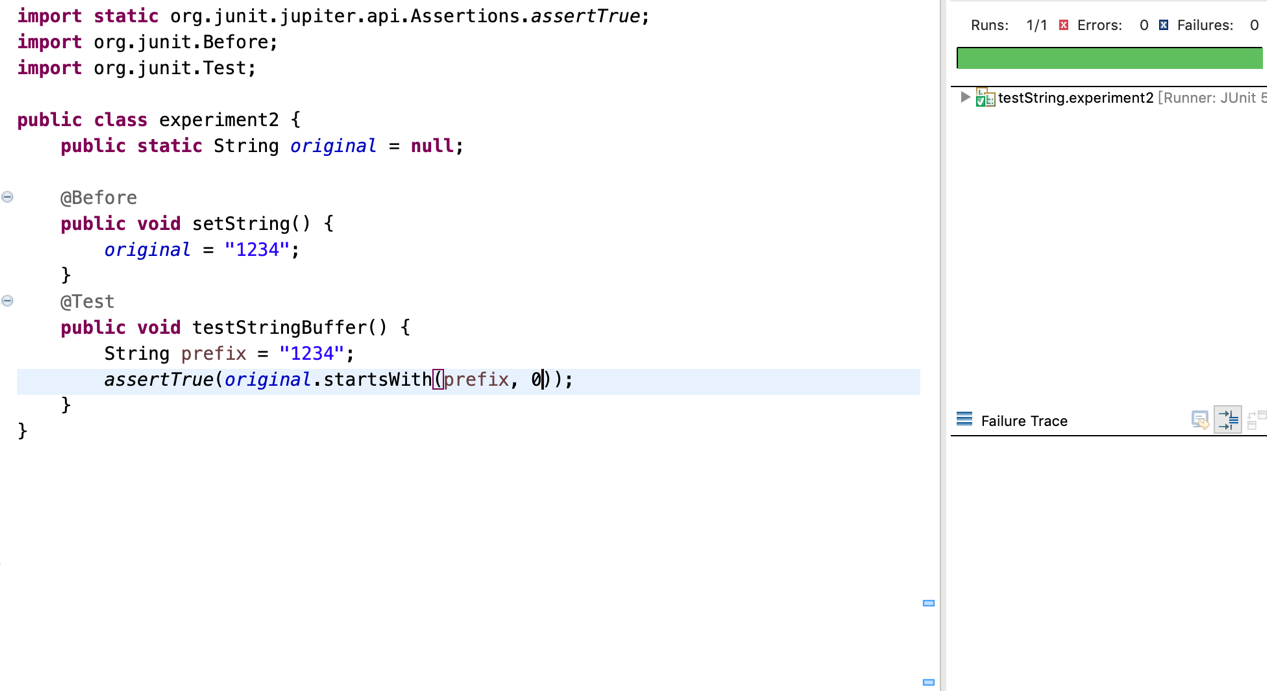


1. original存在prefix子串

构造一个”1234”的original字符串和一个”1234”的prefix字符串，toffset为2

使用@Before 延迟加载构造的字符串

assertTrue测试后返回true



1. 测试 public String[] split(String regex, int limit)