SOTorrent

MSR Mining Challenge 2019

Sebastian Baltes
@s_baltes

MSR 2019

16th International Conference on
Mining Software Repositories (**MSR 2019**)
May 26-27, Montreal, Canada

empirical-software.engineering

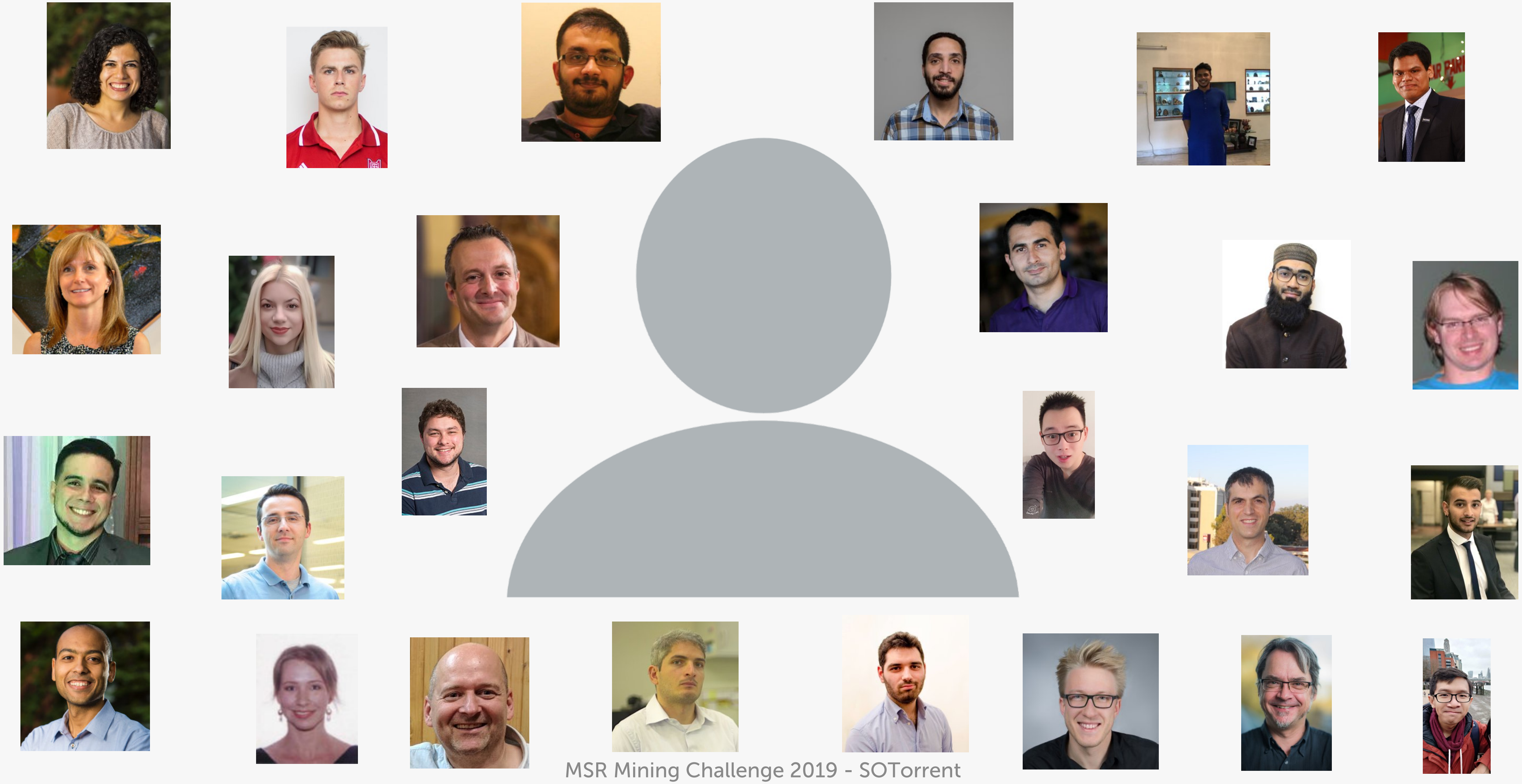
Thanks to my co-chairs!

Christoph Treude

Stephan Diehl

Thanks to all PC members!

Thanks to all authors!



MSR Mining Challenge 2019: Submissions

- 38 abstracts and 27 papers submitted
- 14 papers accepted (52% acceptance rate)
- Diverse range of topics covered, including
  - Quality of code snippets
  - Analysis of duplicate content
  - Evolution of content on Stack Overflow

MSR Mining Challenge 2019: Changes

- Presentation award → **Student presentation award**
- New **open science policy:**
  - Authors are encouraged to **share** their scripts and data in a **replication package**
  - Replication packages should be published on a **preserved archive** such as Figshare or Zenodo to receive a DOI
  - Cite replication package in paper using DOI
  - Please remember that GitHub is not a preserved archive (but repos can be automatically archived on Zenodo)
  - Only papers adhering to the open science policy qualify for the **best paper award**

MSR Mining Challenge 2019: Impact of Changes

- Mining Challenge 2018 (13 accepted papers)
  - 7 (54%) papers did not share any software or data
  - 4 (31%) papers provided links to non-preserved archives, of which 2 were already dead (as of September 2018)
  - **2 (15%) papers provided replication packages on preserved archives**

- Mining Challenge 2019 (14 accepted papers)
  - 5 (36%) papers did not share any software or data
  - 0 (0%) papers provided links to non-preserved archives
  - **9 (64%) papers provided replication packages on preserved archives**
    (8 Zenodo, 1 Figshare)

MSR Mining Challenge 2019: Program

1. SOTorrent: Studying the Origin, Evolution, and Usage of Stack Overflow Code Snippets
2. Mining Rule Violations in JavaScript Code Snippets
3. Snakes in Paradise?: Insecure Python-related Coding Practices in Stack Overflow
4. Man vs Machine - A Study into language identification of Stackoverflow code snippets
5. Python Coding Style Compliance on Stack Overflow

Properties of Snippets

6. Towards Mining Answer Edits to Extract Evolution Patterns in Stack Overflow
7. Analyzing Comment-induced Updates on Stack Overflow
8. What Edits Are Done on Highly Answered Stack Overflow Questions? An Empirical Study

Content Evolution

9. Can Duplicate Posts on Stack Overflow Benefit the Software Development Community?
10. How Often and What StackOverflow Posts Do Developers Reference in Their GitHub Projects?
11. Characterizing Duplicate Code Snippets between Stack Overflow and Tutorials

Clones and References

12. Challenges with Responding to Static Analysis Tool Alerts
13. Impact of stack overflow code snippets on software cohesion: a preliminary study
14. We Need to Talk about Microservices: an Analysis from the Discussions on StackOverflow
15. What do developers know about machine learning: a study of ML discussions on StackOverflow

Other

MSR Mining Challenge 2019: Program

1. **SOTorrent: Studying the Origin, Evolution, and Usage of Stack Overflow Code Snippets**
2. Mining Rule Violations in JavaScript Code Snippets
3. Snakes in Paradise?: Insecure Python-related Coding Practices in Stack Overflow
4. Man vs Machine - A Study into language identification of Stackoverflow code snippets
5. Python Coding Style Compliance on Stack Overflow
6. Towards Mining Answer Edits to Extract Evolution Patterns in Stack Overflow
7. Analyzing Comment-induced Updates on Stack Overflow
8. What Edits Are Done on Highly Answered Stack Overflow Questions? An Empirical Study
9. Can Duplicate Posts on Stack Overflow Benefit the Software Development Community?
10. How Often and What StackOverflow Posts Do Developers Reference in Their GitHub Projects?
11. Characterizing Duplicate Code Snippets between Stack Overflow and Tutorials
12. Challenges with Responding to Static Analysis Tool Alerts
13. Impact of stack overflow code snippets on software cohesion: a preliminary study
14. We Need to Talk about Microservices: an Analysis from the Discussions on StackOverflow
15. What do developers know about machine learning: a study of ML discussions on StackOverflow

SOTorrent

Studying the Origin, Evolution, and Usage of Stack Overflow Code Snippets

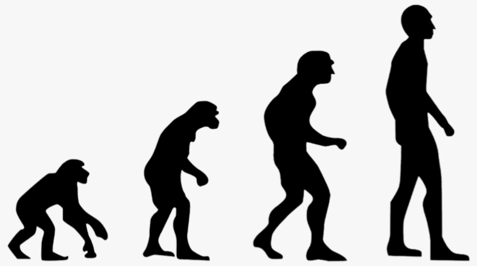
Sebastian Baltes
@s_baltes

empirical-software.engineering

16th International Conference on
Mining Software Repositories (**MSR 2019**)
May 26-27, Montreal, Canada

Why Reconstruct and Analyze SO Post Evolution?

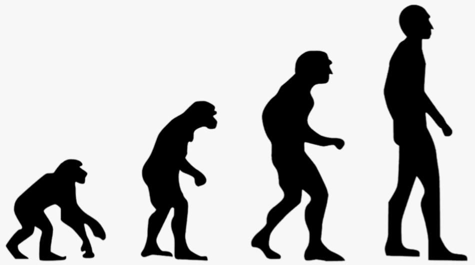
- The content of **14.8 million posts** has been **edited** after creation (SOTorrent 2019-03-17)
- Like other **software artifacts**, SO posts **evolve over time**:
  - Bugs in code snippets are fixed
  - Clarifications are added in text documenting the code
  - Snippets are updated to new language/library versions
- **Copying code** from Stack Overflow (SO) is common, despite licensing, security, and maintainability implications



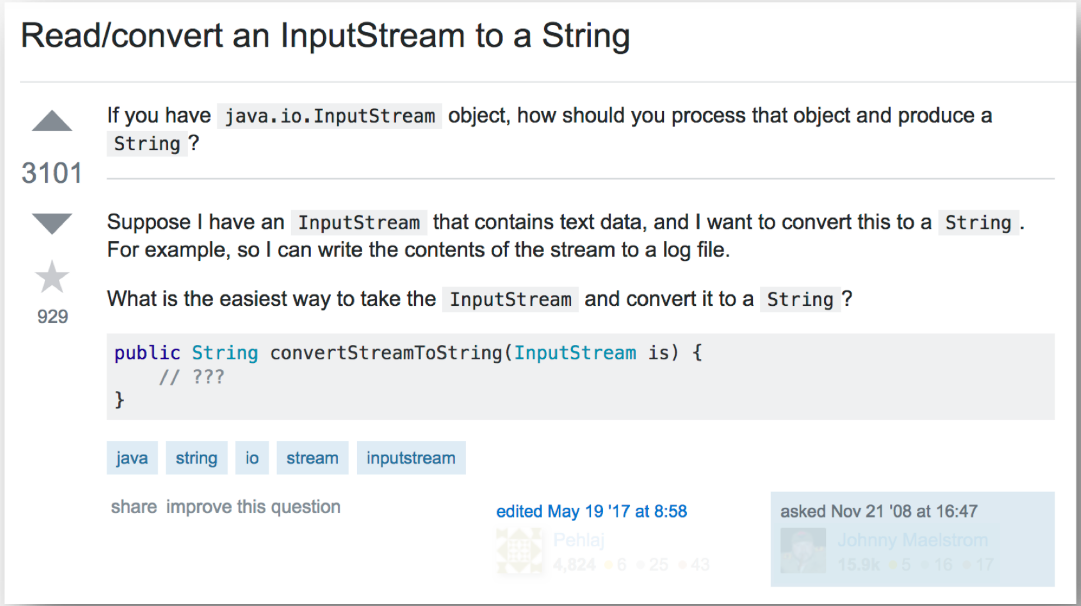
Why Reconstruct and Analyze SO Post Evolution?

Evolution of code on SO differs from regular software projects:

- **Short** code snippets (12 LOC on average)
- **No bug tracking** system (just comments and new answers)
- **No versioning** for individual snippets (just whole posts)



Example



Read/convert an InputStream to a String

3101

929

If you have `java.io.InputStream` object, how should you process that object and produce a `String` ?

Suppose I have an `InputStream` that contains text data, and I want to convert this to a `String`. For example, so I can write the contents of the stream to a log file.

What is the easiest way to take the `InputStream` and convert it to a `String` ?

```java
public String convertStreamToString(InputStream is) {
    // ???
}
```

java string io stream inputstream

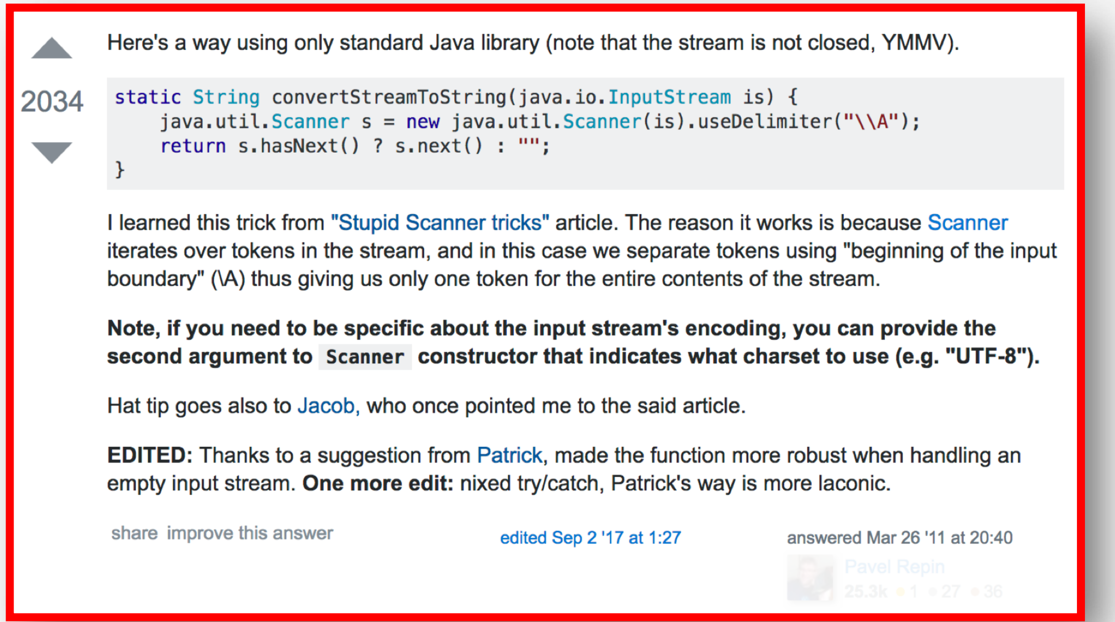
share improve this question

edited May 19 '17 at 8:58
Pehlaj
4,824 ●6 ●25 ●43

asked Nov 21 '08 at 16:47
Johnny Maelstrom
15.9k ●5 ●16 ●17

Question

https://stackoverflow.com/q/309424



2034

Here's a way using only standard Java library (note that the stream is not closed, YMMV).

```java
static String convertStreamToString(java.io.InputStream is) {
    java.util.Scanner s = new java.util.Scanner(is).useDelimiter("\\A");
    return s.hasNext() ? s.next() : "";
}
```

I learned this trick from "Stupid Scanner tricks" article. The reason it works is because Scanner iterates over tokens in the stream, and in this case we separate tokens using "beginning of the input boundary" (\A) thus giving us only one token for the entire contents of the stream.

Note, if you need to be specific about the input stream's encoding, you can provide the second argument to `Scanner` constructor that indicates what charset to use (e.g. "UTF-8").

Hat tip goes also to Jacob, who once pointed me to the said article.

EDITED: Thanks to a suggestion from Patrick, made the function more robust when handling an empty input stream. **One more edit:** nixed try/catch, Patrick's way is more laconic.

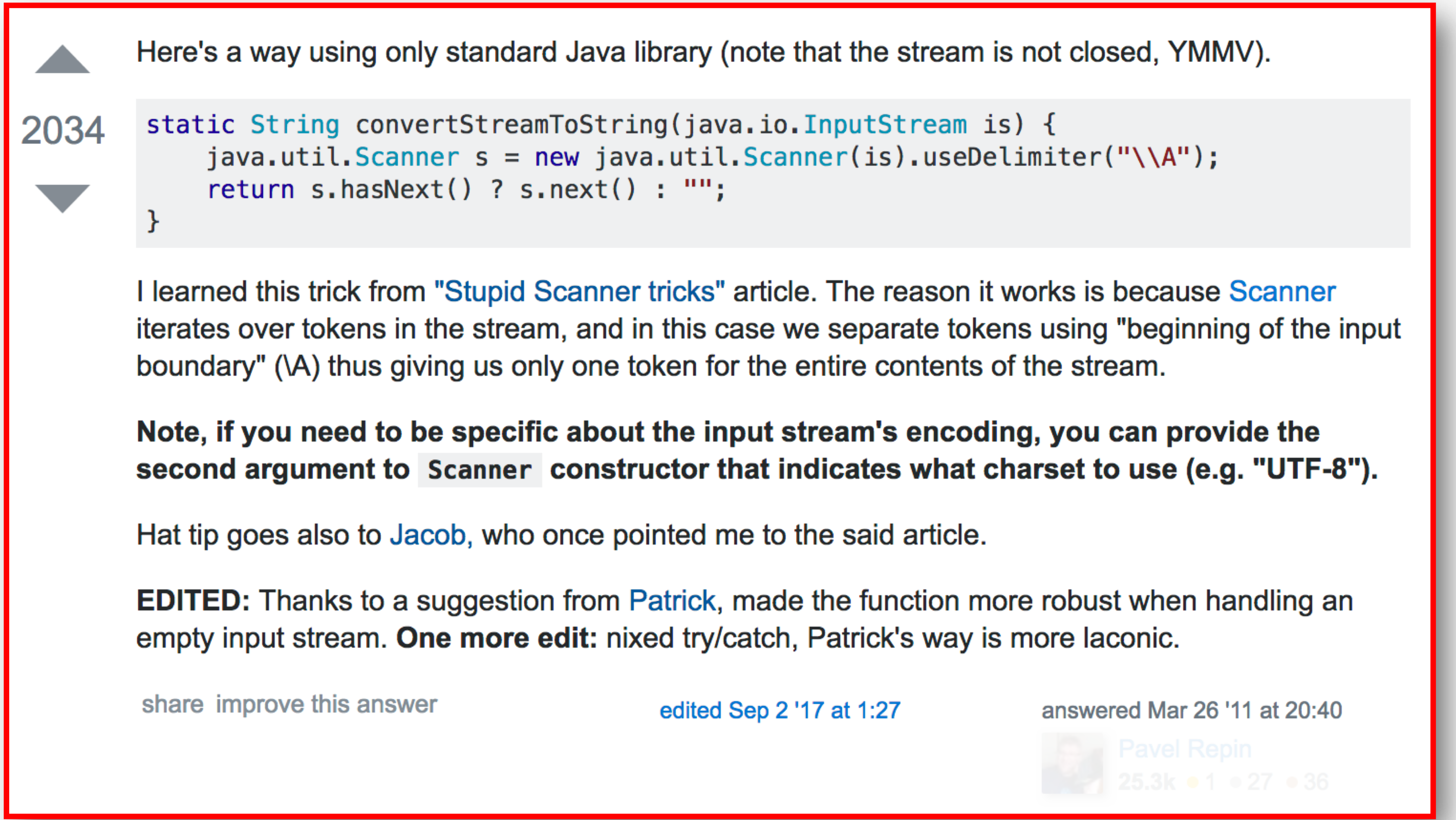
share improve this answer

edited Sep 2 '17 at 1:27

answered Mar 26 '11 at 20:40
Pavel Repin
25.3k ●1 ●27 ●36

Answer

https://stackoverflow.com/a/5445161


2034

Here's a way using only standard Java library (note that the stream is not closed, YMMV).

```
static String convertStreamToString(java.io.InputStream is) {
    java.util.Scanner s = new java.util.Scanner(is).useDelimiter("\\A");
    return s.hasNext() ? s.next() : "";
}
```

I learned this trick from "Stupid Scanner tricks" article. The reason it works is because Scanner iterates over tokens in the stream, and in this case we separate tokens using "beginning of the input boundary" (\A) thus giving us only one token for the entire contents of the stream.

Note, if you need to be specific about the input stream's encoding, you can provide the second argument to `Scanner` constructor that indicates what charset to use (e.g. "UTF-8").

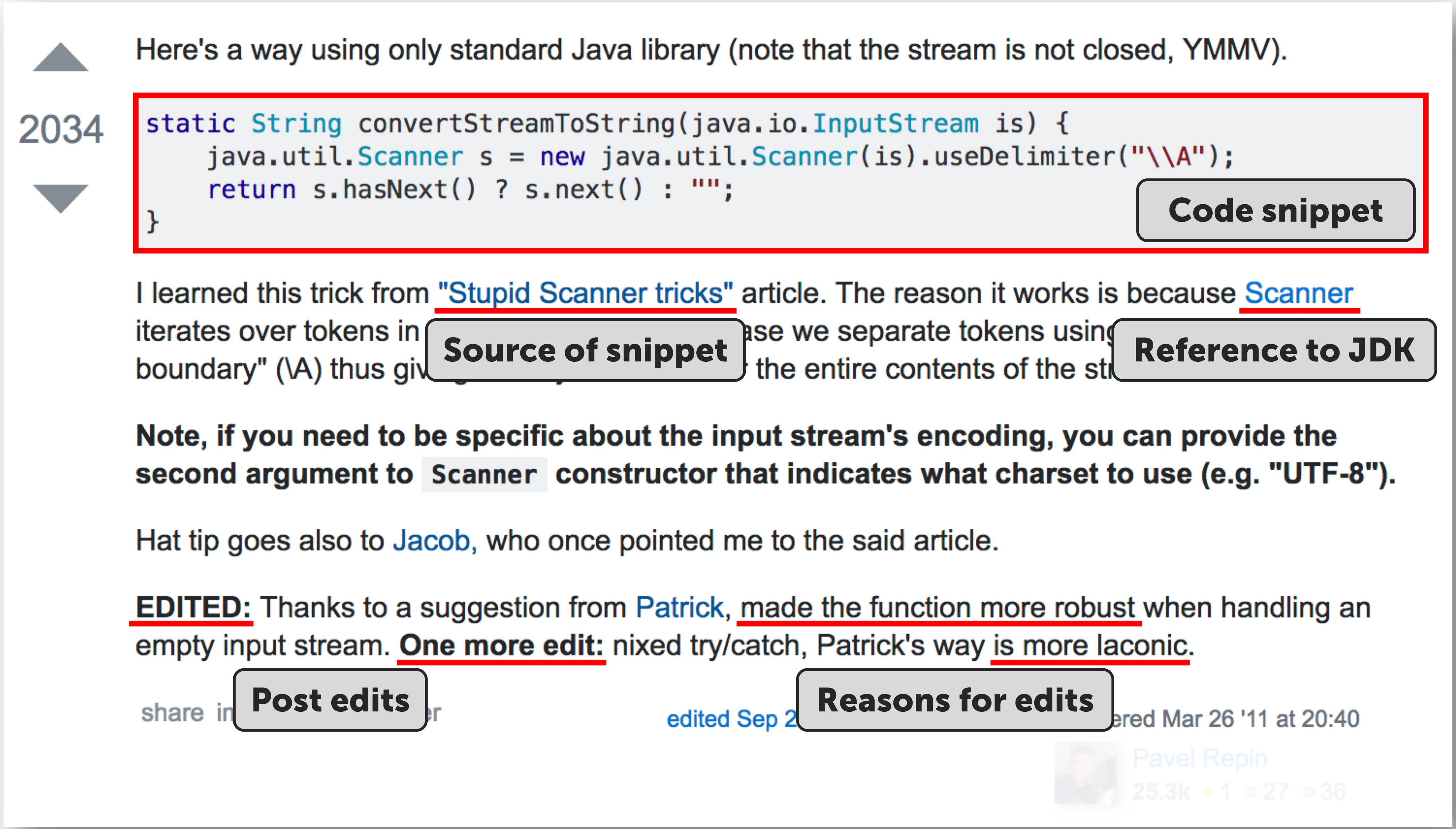
Hat tip goes also to Jacob, who once pointed me to the said article.

EDITED: Thanks to a suggestion from Patrick, made the function more robust when handling an empty input stream. **One more edit:** nixed try/catch, Patrick's way is more laconic.

share improve this answer   edited Sep 2 '17 at 1:27   answered Mar 26 '11 at 20:40

Pavel Repin
25.3k 1 27 36

https://stackoverflow.com/a/5445161



2034

Here's a way using only standard Java library (note that the stream is not closed, YMMV).

```
static String convertStreamToString(java.io.InputStream is) {
    java.util.Scanner s = new java.util.Scanner(is).useDelimiter("\\A");
    return s.hasNext() ? s.next() : "";
}
```

I learned this trick from "Stupid Scanner tricks" article. The reason it works is because Scanner iterates over tokens in ase we separate tokens using boundary" (\A) thus giv the entire contents of the st

Note, if you need to be specific about the input stream's encoding, you can provide the second argument to `Scanner` constructor that indicates what charset to use (e.g. "UTF-8").

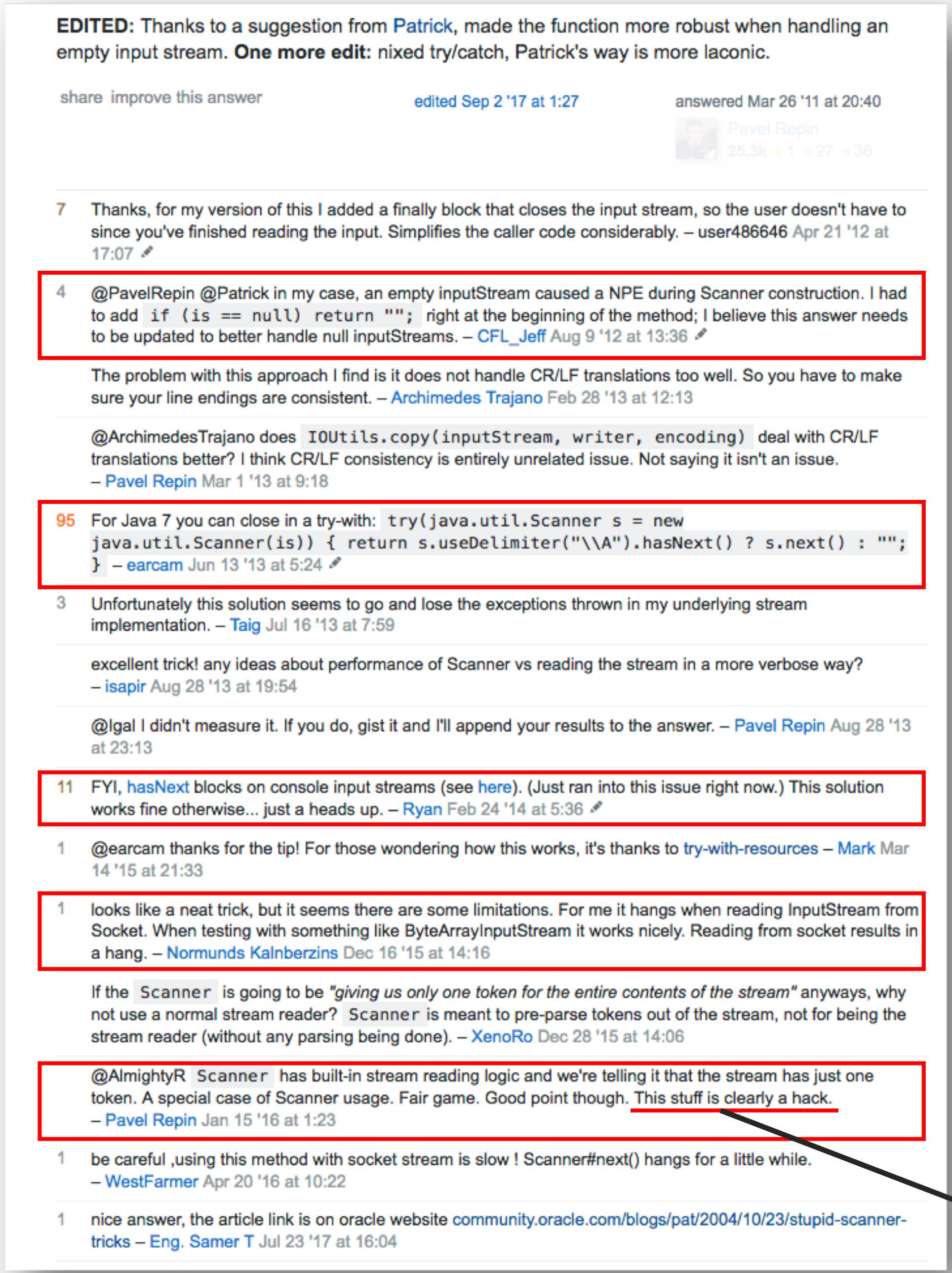
Hat tip goes also to Jacob, who once pointed me to the said article.

EDITED: Thanks to a suggestion from Patrick, made the function more robust when handling an empty input stream. **One more edit:** nixed try/catch, Patrick's way is more laconic.

share in er edited Sep 2 ered Mar 26 '11 at 20:40

Pavel Repin
25.3k ● 1 ● 27 ● 36

Comments



EDITED: Thanks to a suggestion from Patrick, made the function more robust when handling an empty input stream. **One more edit:** nixed try/catch, Patrick's way is more laconic.

share improve this answer edited Sep 2 '17 at 1:27 answered Mar 26 '11 at 20:40

Pavel Repin
25.3k ● 1 ● 27 ● 36

7 Thanks, for my version of this I added a finally block that closes the input stream, so the user doesn't have to since you've finished reading the input. Simplifies the caller code considerably. – user486646 Apr 21 '12 at 17:07

4 @PavelRepin @Patrick in my case, an empty inputStream caused a NPE during Scanner construction. I had to add `if (is == null) return "";` right at the beginning of the method; I believe this answer needs to be updated to better handle null inputStreams. – CFL_Jeff Aug 9 '12 at 13:36

The problem with this approach I find is it does not handle CR/LF translations too well. So you have to make sure your line endings are consistent. – Archimedes Trajano Feb 28 '13 at 12:13

@ArchimedesTrajano does `IOUtils.copy(inputStream, writer, encoding)` deal with CR/LF translations better? I think CR/LF consistency is entirely unrelated issue. Not saying it isn't an issue. – Pavel Repin Mar 1 '13 at 9:18

95 For Java 7 you can close in a try-with: `try(java.util.Scanner s = new java.util.Scanner(is)) { return s.useDelimiter("\\A").hasNext() ? s.next() : ""; }` – earcam Jun 13 '13 at 5:24

3 Unfortunately this solution seems to go and lose the exceptions thrown in my underlying stream implementation. – Taig Jul 16 '13 at 7:59

excellent trick! any ideas about performance of Scanner vs reading the stream in a more verbose way? – isapir Aug 28 '13 at 19:54

@Igal I didn't measure it. If you do, gist it and I'll append your results to the answer. – Pavel Repin Aug 28 '13 at 23:13

11 FYI, hasNext blocks on console input streams (see here). (Just ran into this issue right now.) This solution works fine otherwise... just a heads up. – Ryan Feb 24 '14 at 5:36

1 @earcam thanks for the tip! For those wondering how this works, it's thanks to try-with-resources – Mark Mar 14 '15 at 21:33

1 looks like a neat trick, but it seems there are some limitations. For me it hangs when reading InputStream from Socket. When testing with something like ByteArrayInputStream it works nicely. Reading from socket results in a hang. – Normunds Kalnberzins Dec 16 '15 at 14:16

If the `Scanner` is going to be *"giving us only one token for the entire contents of the stream"* anyways, why not use a normal stream reader? `Scanner` is meant to pre-parse tokens out of the stream, not for being the stream reader (without any parsing being done). – XenoRo Dec 28 '15 at 14:06

@AlmightyR `Scanner` has built-in stream reading logic and we're telling it that the stream has just one token. A special case of Scanner usage. Fair game. Good point though. This stuff is clearly a hack. – Pavel Repin Jan 15 '16 at 1:23

1 be careful ,using this method with socket stream is slow ! Scanner#next() hangs for a little while. – WestFarmer Apr 20 '16 at 10:22

1 nice answer, the article link is on oracle website community.oracle.com/blogs/pat/2004/10/23/stupid-scanner-tricks – Eng. Samer T Jul 23 '17 at 16:04

Bug report

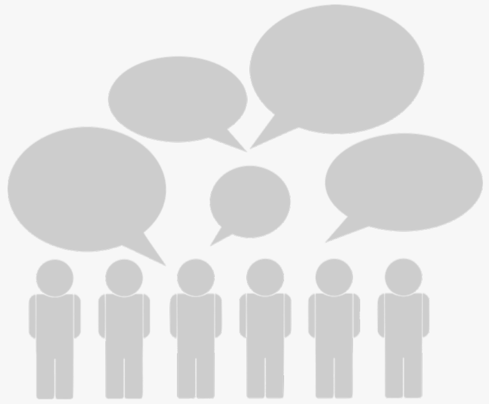
Alternative solution

Bug report

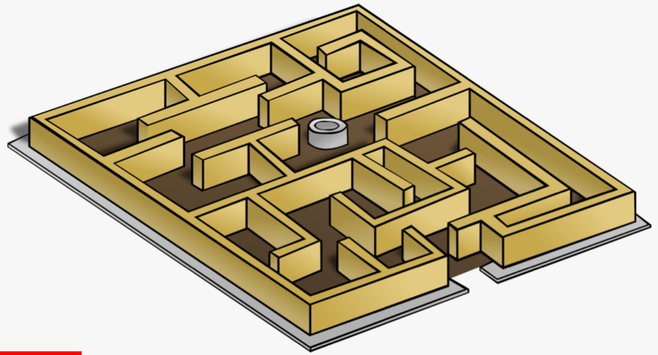
Bug report

Comment by author

This stuff is clearly a hack.



Even for such a simple code snippet, the **context** is quite **complex**:

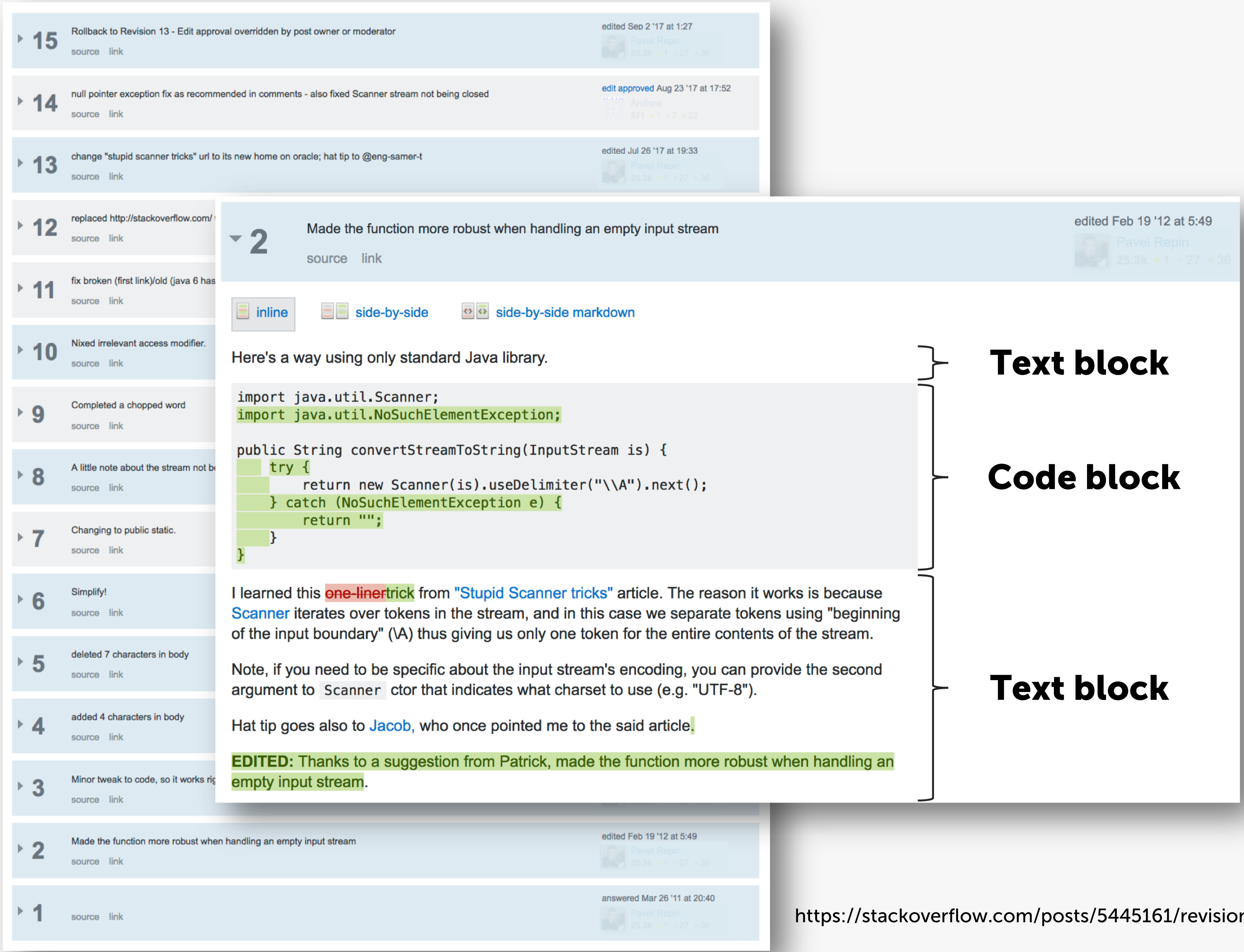


- The snippet is based on an **external source**
- Hidden in the **comments**, the author acknowledges that his solution is *"clearly a hack"*
- There are several **bug reports** pointing to issues
- Has the snippet been **edited** to fix those issues?
- Is the snippet **safe** to use?

SO Revisions

Problems:

- Version history is only available on the level of whole posts, thus **individual code snippets hard to trace**
- **Comments and edits** are not linked
- Not visible how **external sources** are related



15 Rollback to Revision 13 - Edit approval overridden by post owner or moderator
source link
edited Sep 2 '17 at 1:27
Pavel Repin

14 null pointer exception fix as recommended in comments - also fixed Scanner stream not being closed
source link
edit approved Aug 23 '17 at 17:52
Andrew

13 change "stupid scanner tricks" url to its new home on oracle; hat tip to @eng-samer-t
source link
edited Jul 26 '17 at 19:33
Pavel Repin

12 replaced http://stackoverflow.com/
source link

11 fix broken (first link)/old (java 6 has
source link

10 Nixed irrelevant access modifier.
source link

9 Completed a chopped word
source link

8 A little note about the stream not b
source link

7 Changing to public static.
source link

6 Simplify!
source link

5 deleted 7 characters in body
source link

4 added 4 characters in body
source link

3 Minor tweak to code, so it works ri
source link

2 Made the function more robust when handling an empty input stream
source link
edited Feb 19 '12 at 5:49

1
source link
answered Mar 26 '11 at 20:40

2 Made the function more robust when handling an empty input stream
source link
edited Feb 19 '12 at 5:49
Pavel Repin

inline side-by-side side-by-side markdown

Here's a way using only standard Java library.

```
import java.util.Scanner;
import java.util.NoSuchElementException;

public String convertStreamToString(InputStream is) {
    try {
        return new Scanner(is).useDelimiter("\\A").next();
    } catch (NoSuchElementException e) {
        return "";
    }
}
```

I learned this ~~one-liner~~trick from "Stupid Scanner tricks" article. The reason it works is because Scanner iterates over tokens in the stream, and in this case we separate tokens using "beginning of the input boundary" (\A) thus giving us only one token for the entire contents of the stream.

Note, if you need to be specific about the input stream's encoding, you can provide the second argument to `Scanner` ctor that indicates what charset to use (e.g. "UTF-8").

Hat tip goes also to Jacob, who once pointed me to the said article.

EDITED: Thanks to a suggestion from Patrick, made the function more robust when handling an empty input stream.

https://stackoverflow.com/posts/5445161/revisions

SOTorrent

SOTorrent

Open dataset based on the official Stack Overflow data dump

Problem:

- Version history only available on the level of whole posts
- Analysis of individual code snippets
- Relation to external resources

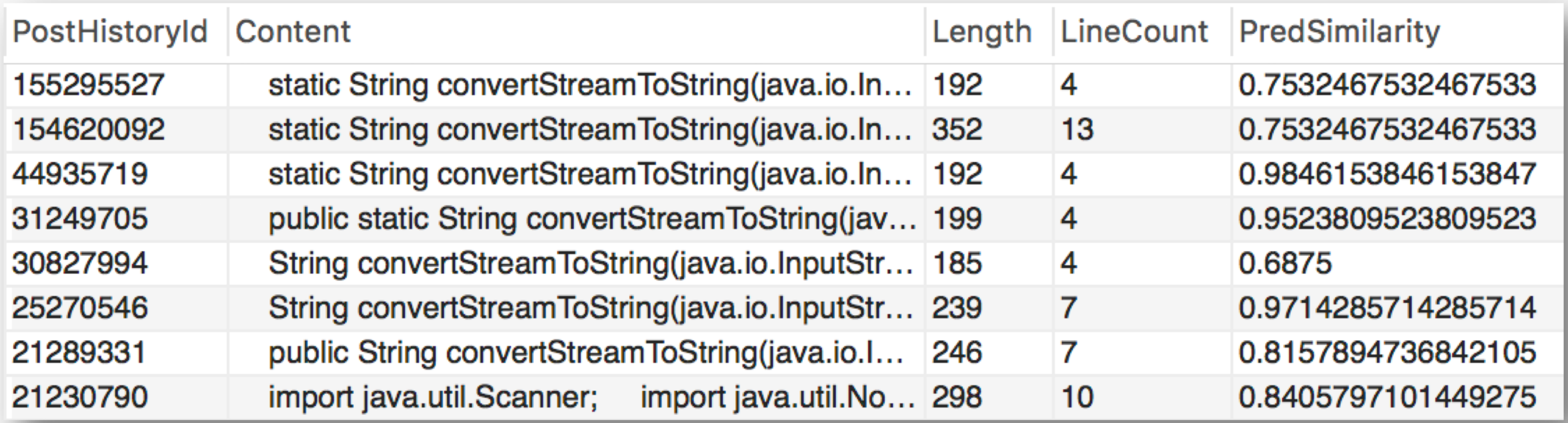
SOTorrent:

- Reconstructed evolution of individual post blocks ✓
- Possible ✓
- Supports researchers in analyzing links from/to posts ✓

Retrieve all versions of a code snippet:

```
SELECT PostHistoryId, Content, Length, LineCount, PredSimilarity
FROM PostBlockVersion
WHERE PostId=5445161 AND LocalId=2 AND PredEqual=0
ORDER BY PostHistoryId DESC;
```

Most recent version

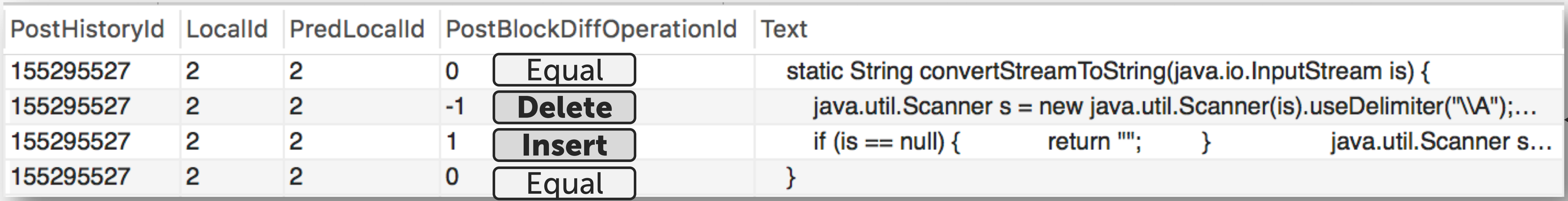


PostHistoryId	Content	Length	LineCount	PredSimilarity
155295527	static String convertStreamToString(java.io.In…	192	4	0.7532467532467533
154620092	static String convertStreamToString(java.io.In…	352	13	0.7532467532467533
44935719	static String convertStreamToString(java.io.In…	192	4	0.9846153846153847
31249705	public static String convertStreamToString(jav…	199	4	0.9523809523809523
30827994	String convertStreamToString(java.io.InputStr…	185	4	0.6875
25270546	String convertStreamToString(java.io.InputStr…	239	7	0.9714285714285714
21289331	public String convertStreamToString(java.io.I…	246	7	0.8157894736842105
21230790	import java.util.Scanner; import java.util.No…	298	10	0.8405797101449275

Retrieve line-based difference for latest version:

```sql
SELECT PostHistoryId, LocalId, PredLocalId, PostBlockDiffOperationId, Text
FROM PostBlockDiff
WHERE PostHistoryId=155295527 AND LocalId=2;
```

Changed lines

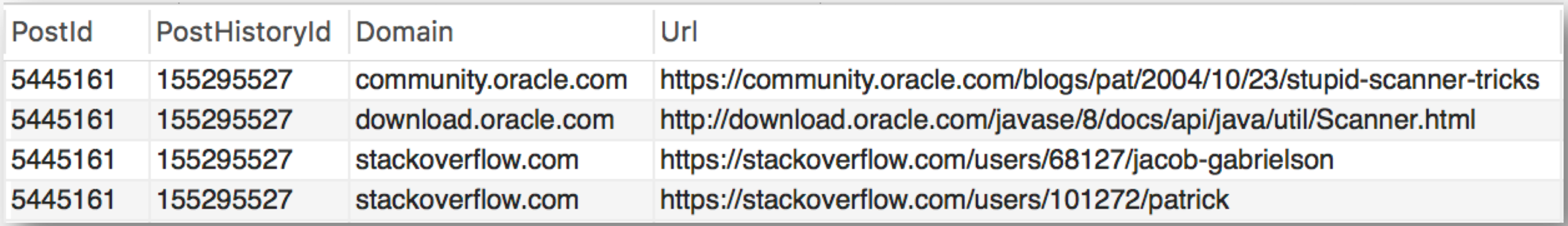


PostHistoryId	LocalId	PredLocalId	PostBlockDiffOperationId	Text
155295527	2	2	0 Equal	static String convertStreamToString(java.io.InputStream is) {
155295527	2	2	-1 **Delete**	java.util.Scanner s = new java.util.Scanner(is).useDelimiter("\\A");...
155295527	2	2	1 **Insert**	if (is == null) { return ""; } java.util.Scanner s...
155295527	2	2	0 Equal	}

Retrieve links from a post version:

```sql
SELECT PostId, PostHistoryId, Domain, Url
FROM PostVersionUrl
WHERE PostHistoryId=155295527;
```

⇩

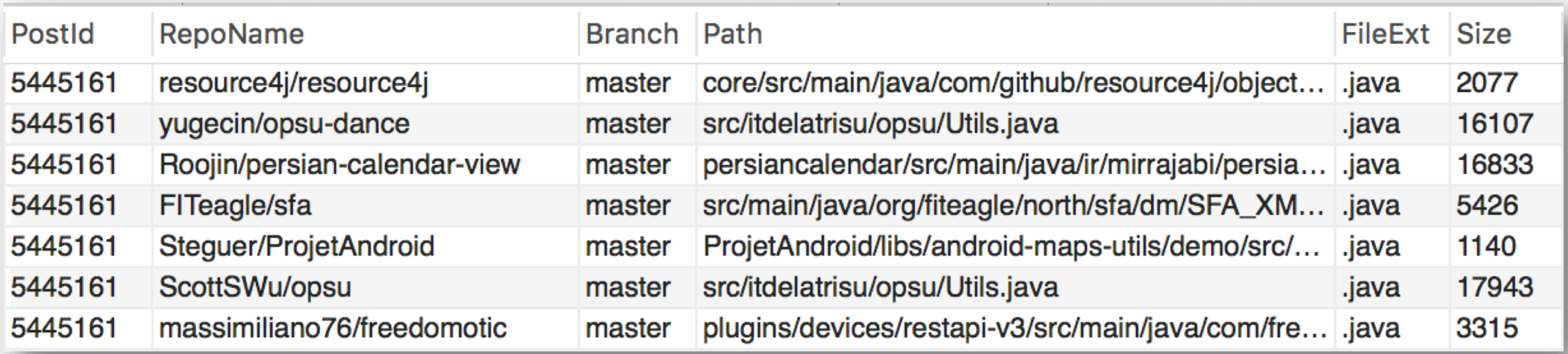


PostId	PostHistoryId	Domain	Url
5445161	155295527	community.oracle.com	https://community.oracle.com/blogs/pat/2004/10/23/stupid-scanner-tricks
5445161	155295527	download.oracle.com	http://download.oracle.com/javase/8/docs/api/java/util/Scanner.html
5445161	155295527	stackoverflow.com	https://stackoverflow.com/users/68127/jacob-gabrielson
5445161	155295527	stackoverflow.com	https://stackoverflow.com/users/101272/patrick

Retrieve links from GitHub repos to post:

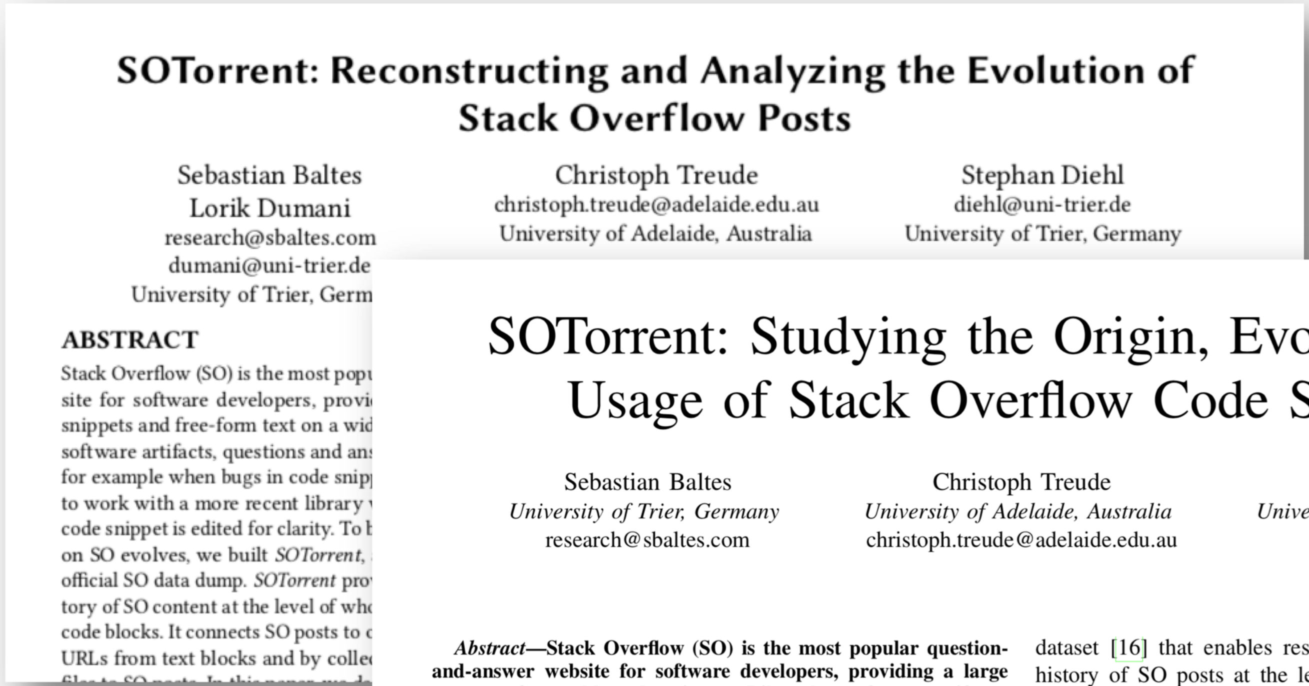
```
SELECT PostId, RepoName, Branch, Path, FileExt, Size, Copies
FROM PostReferenceGH
WHERE PostId=5445161;
```

Referenced in 113 distinct repos



PostId	RepoName	Branch	Path	FileExt	Size
5445161	resource4j/resource4j	master	core/src/main/java/com/github/resource4j/object…	.java	2077
5445161	yugecin/opsu-dance	master	src/itdelatrisu/opsu/Utils.java	.java	16107
5445161	Roojin/persian-calendar-view	master	persiancalendar/src/main/java/ir/mirrajabi/persia…	.java	16833
5445161	FITeagle/sfa	master	src/main/java/org/fiteagle/north/sfa/dm/SFA_XM…	.java	5426
5445161	Steguer/ProjetAndroid	master	ProjetAndroid/libs/android-maps-utils/demo/src/…	.java	1140
5445161	ScottSWu/opsu	master	src/itdelatrisu/opsu/Utils.java	.java	17943
5445161	massimiliano76/freedomotic	master	plugins/devices/restapi-v3/src/main/java/com/fre…	.java	3315

…



SOTorrent: Reconstructing and Analyzing the Evolution of Stack Overflow Posts

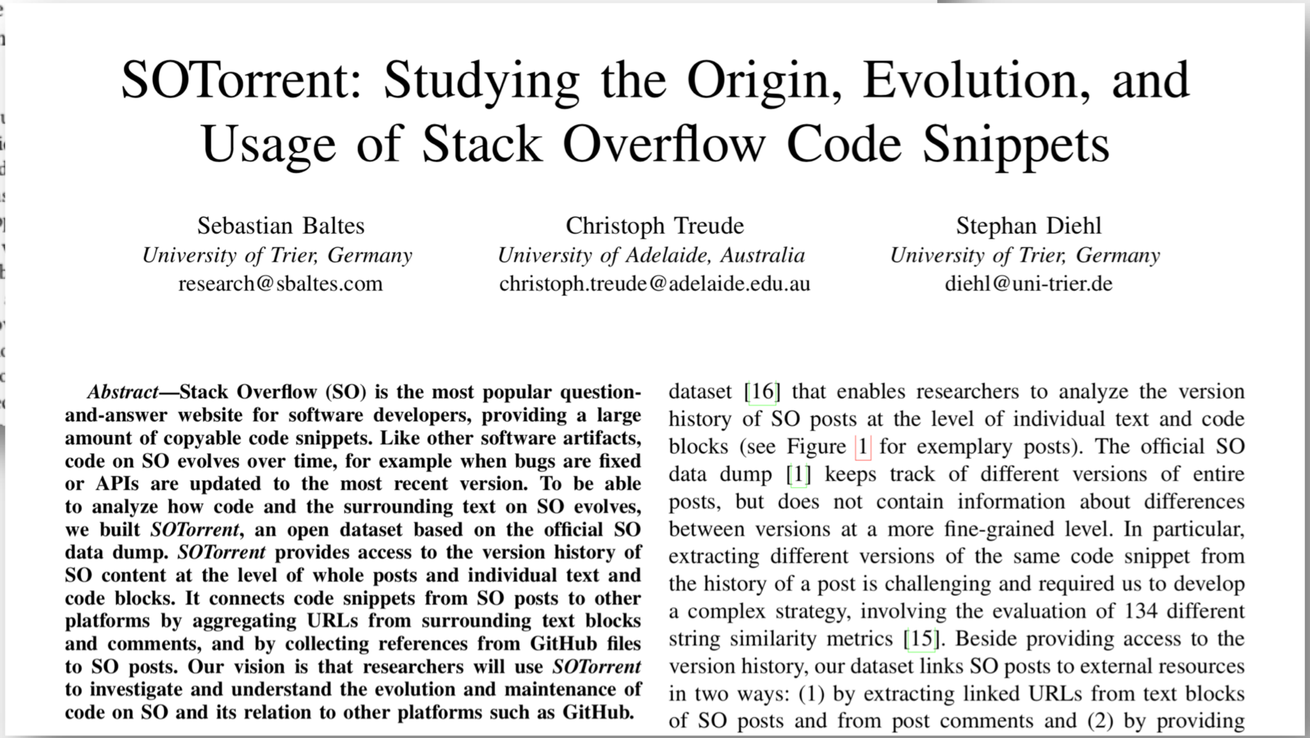
Sebastian Baltes
Lorik Dumani
research@sbaltes.com
dumani@uni-trier.de
University of Trier, Germ

Christoph Treude
christoph.treude@adelaide.edu.au
University of Adelaide, Australia

Stephan Diehl
diehl@uni-trier.de
University of Trier, Germany

ABSTRACT

Stack Overflow (SO) is the most popu
site for software developers, provi
snippets and free-form text on a wid
software artifacts, questions and an
for example when bugs in code snip
to work with a more recent library
code snippet is edited for clarity. To
on SO evolves, we built *SOTorrent*,
official SO data dump. *SOTorrent* pro
tory of SO content at the level of wh
code blocks. It connects SO posts to
URLs from text blocks and by colle



SOTorrent: Studying the Origin, Evolution, and Usage of Stack Overflow Code Snippets

Sebastian Baltes
University of Trier, Germany
research@sbaltes.com

Christoph Treude
University of Adelaide, Australia
christoph.treude@adelaide.edu.au

Stephan Diehl
University of Trier, Germany
diehl@uni-trier.de

***Abstract*—Stack Overflow (SO) is the most popular question-and-answer website for software developers, providing a large amount of copyable code snippets. Like other software artifacts, code on SO evolves over time, for example when bugs are fixed or APIs are updated to the most recent version. To be able to analyze how code and the surrounding text on SO evolves, we built *SOTorrent*, an open dataset based on the official SO data dump. *SOTorrent* provides access to the version history of SO content at the level of whole posts and individual text and code blocks. It connects code snippets from SO posts to other platforms by aggregating URLs from surrounding text blocks and comments, and by collecting references from GitHub files to SO posts. Our vision is that researchers will use *SOTorrent* to investigate and understand the evolution and maintenance of code on SO and its relation to other platforms such as GitHub.**

dataset [16] that enables researchers to analyze the version history of SO posts at the level of individual text and code blocks (see Figure 1 for exemplary posts). The official SO data dump [1] keeps track of different versions of entire posts, but does not contain information about differences between versions at a more fine-grained level. In particular, extracting different versions of the same code snippet from the history of a post is challenging and required us to develop a complex strategy, involving the evaluation of 134 different string similarity metrics [15]. Beside providing access to the version history, our dataset links SO posts to external resources in two ways: (1) by extracting linked URLs from text blocks of SO posts and from post comments and (2) by providing

MSR 2018/19

sotorrent.org

Open Data

Dataset available on Zenodo and BigQuery

Cloud Icon CC BY 3.0 smashicons on flaticon.com

Voting for Best Student Presentation

MSR Mining Challenge 2019: Program

1. SOTorrent: Studying the Origin, Evolution, and Usage of Stack Overflow Code Snippets
2. Mining Rule Violations in JavaScript Code Snippets
3. Snakes in Paradise?: Insecure Python-related Coding Practices in Stack
4. Man vs Machine - A Study into language identification of Stackoverflov
5. Python Coding Style Compliance on Stack Overflow
6. Towards Mining Answer Edits to Extract Evolution Patterns in Stack Ov
7. Analyzing Comment-induced Updates on Stack Overflow
8. What Edits Are Done on Highly Answered Stack Overflow Questions? A
9. Can Duplicate Posts on Stack Overflow Benefit the Software Developm
10. How Often and What StackOverflow Posts Do Developers Reference in Their GitHub Projects?
11. Characterizing Duplicate Code Snippets between Stack Overflow and Tutorials
12. Challenges with Responding to Static Analysis Tool Alerts
13. Impact of stack overflow code snippets on software cohesion: a preliminary study
14. We Need to Talk about Microservices: an Analysis from the Discussions on StackOverflow
15. What do developers know about machine learning: a study of ML discussions on StackOverflow

I vote for paper number

$\boxed{1}$

to receive the best student presentation award.

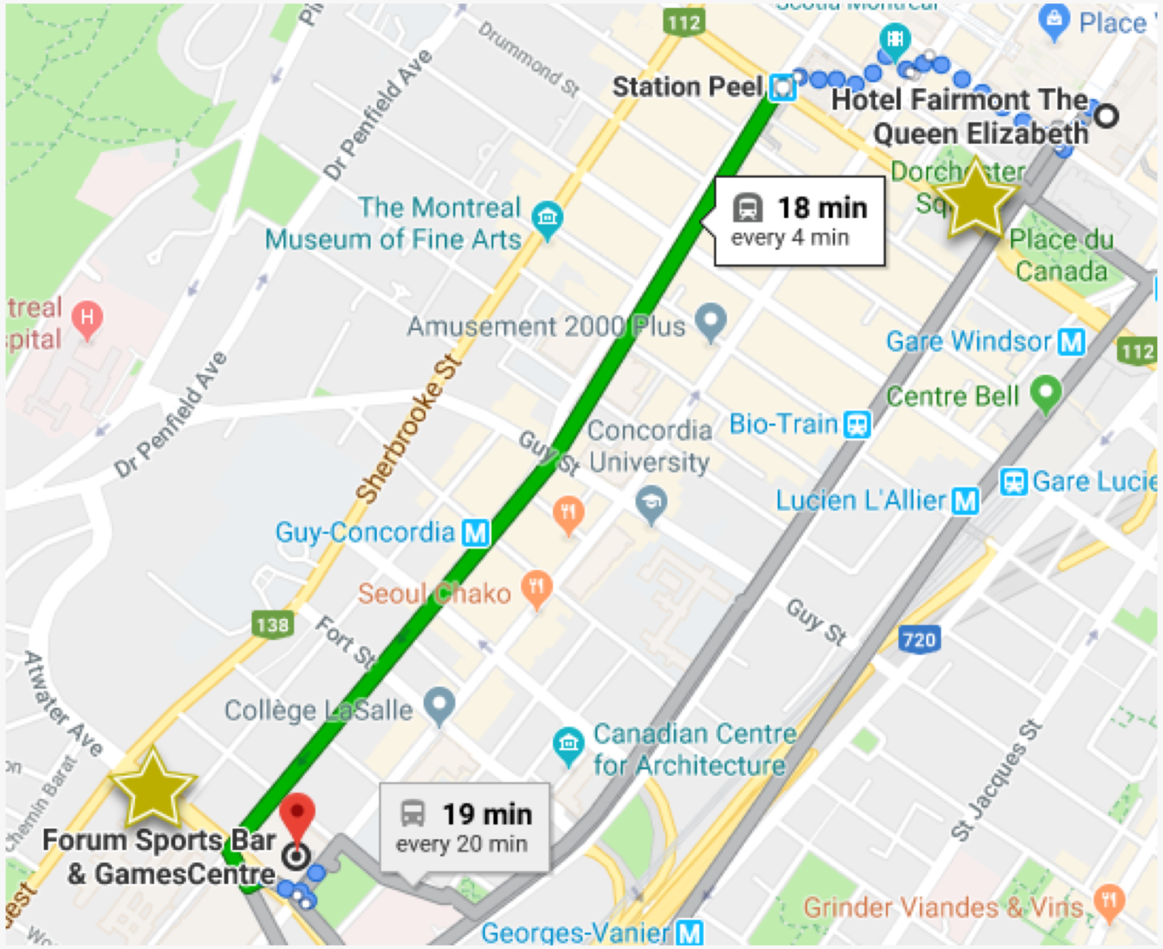
MSR 2019

MSR Mining Challenge 2019: Program

1. SOTorrent: Studying the Origin, Evolution, and Usage of Stack Overflow Code Snippets
2. **Mining Rule Violations in JavaScript Code Snippets**
3. Snakes in Paradise?: Insecure Python-related Coding Practices in Stack Overflow
4. Man vs Machine - A Study into language identification of Stackoverflow code snippets
5. Python Coding Style Compliance on Stack Overflow
6. Towards Mining Answer Edits to Extract Evolution Patterns in Stack Overflow
7. Analyzing Comment-induced Updates on Stack Overflow
8. What Edits Are Done on Highly Answered Stack Overflow Questions? An Empirical Study
9. Can Duplicate Posts on Stack Overflow Benefit the Software Development Community?
10. How Often and What StackOverflow Posts Do Developers Reference in Their GitHub Projects?
11. Characterizing Duplicate Code Snippets between Stack Overflow and Tutorials
12. Challenges with Responding to Static Analysis Tool Alerts
13. Impact of stack overflow code snippets on software cohesion: a preliminary study
14. We Need to Talk about Microservices: an Analysis from the Discussions on StackOverflow
15. What do developers know about machine learning: a study of ML discussions on StackOverflow

Voting for Best Student Presentation

Banquet



- Forum de Montreal Games Centre 2313 Rue St-Catherine Ouest
- 3rd floor (escalator/elevator)
- From 18:30 on: drinks, arcades, billiards, air hockey, table tennis!
- From 20:00 on: buffet dinner!
- **You need your MSR dinner voucher to attend!**
- Please email irina@sians.org if you will not attend the dinner!

MSR Mining Challenge 2019: Program

1. SOTorrent: Studying the Origin, Evolution, and Usage of Stack Overflow Code Snippets
2. **Mining Rule Violations in JavaScript Code Snippets (Uriel Campos)**
3. **Snakes in Paradise?: Insecure Python-related Coding Practices in Stack Overflow (Akond Rahman)**
4. Man vs Machine - A Study into language identification of Stackoverflow code snippets
5. **Python Coding Style Compliance on Stack Overflow (Wenjie Boon)**
6. Towards Mining Answer Edits to Extract Evolution Patterns in Stack Overflow
7. Analyzing Comment-induced Updates on Stack Overflow
8. **What Edits Are Done on Highly Answered Stack Overflow Questions? An Empirical Study (Xianhao Jin)**
9. **Can Duplicate Posts on Stack Overflow Benefit the Software Development Community? (Durham Abric)**
10. **How Often and What StackOverflow Posts Do Developers Reference in Their GitHub Projects? (Saraj Singh Manes)**
11. **Characterizing Duplicate Code Snippets between Stack Overflow and Tutorials (Agnieszka Ciborowska)**
12. **Challenges with Responding to Static Analysis Tool Alerts (Nasif Imtiaz)**
13. **Impact of stack overflow code snippets on software cohesion: a preliminary study (Mashal Ahmad)**
14. We Need to Talk about Microservices: an Analysis from the Discussions on StackOverflow
15. **What do developers know about machine learning: a study of ML discussions on StackOverflow (Abdul Ali Bangash)**