



Salutation **First name** Middle na Last name

Contact cl Contact fr Email



KARSTEN SILZ, 21 OCTOBER 2020 JAVERS: CUDE AUDIT LUGS EASILY IN JAVA

Version Details



Version 1



Version 3 by Karsten

Sat, 26 September 2020, 14:38



Version 1 by Joe

Thu, 17 September 2020, 22:14

 $\mathbf{\mathbf{v}}$

2 changes, marked in red.

n	Mr.	Salutation	Mr.
е	Jon	First name	John
me		Middle name	
9	Snow	Last name	Snow
hannel		Contact channel	
requency		Contact frequency	
	Work:	Email	<u>john.snow@nights</u>
	<u>jon.snow@nights-</u>		watch.com
	watch.com		







SAMPLE CODE B APPS TOD





Versio	n Details		🗲 Bac
			Version 1 🗸 👕 Clea
Version Sat, 26 Se	3 by Karsten ptember 2020, 14:38	Ver Thu 2 ch	rsion 1 by Joe , 17 September 2020, 22:14 hanges, marked in red.
Salutation	Mr.	Salutation	Mr.
First name	Jon	First name	John
Middle name		Middle name	
Last name	Snow	Last name	Snow
Contact channel		Contact chann	el
Contact frequency		Contact freque	ency
Email	Work:	Email	john.snow@nights-
	jon.snow@nights-		watch.com
	watch.com		

KARSTEN SILZ, 21 OCTOBER 2020 JAVERS: CODE AUDIT LOGS EASILY IN JAVA

OPTIMZED FOR **NOUTUBEON** A THE SCREEN PHONE 6



SUFFERED FROM CHANGED DATA?





SHOW AUDT LOG TO **USERS: WHO CHANGED** WHAT WHEN HOW?

CUSTOMER: NO EMAIL

EMAIL ADDRESS: OK





5) Version Details





Version 3 by Karsten

Sat, 26 September 2020, 14:38

Salutation
First name
Middle name

Last name

Mr. Jon

Snow

Contact channel Contact frequency Email

Work: jon.snow@nightswatch.com

Compare

Version 1



 \sim



Version 1 by Joe

Thu, 17 September 2020, 22:14

2 changes, marked in red.

Salutation	
First name	
Middle name	
Last name	

Contact channel Contact frequency Email

Mr. John

Snow

john.snow@nightswatch.com



Version Details COMPARE 2 VERSIONS



Version 3 by Karsten

Sat, 26 September 2020, 14:38

Salutation
First name
Middle name

Last name

Mr. Jon

Snow

Contact channel Contact frequency Email

Work: jon.snow@nightswatch.com



Version 1 by Joe

Thu, 17 September 2020, 22:14

2 changes, marked in red.

Salutation First name Middle name

Last name

Contact channel Contact frequency Email Mr. John

Snow

john.snow@nightswatch.com





Version 3 by Karsten Sat, 26 September 2020, 14:38

VERSION INFORMATION



Version 1 by Joe

Thu, 17 September 2020, 22:14

Mr.







Salutation	Mr.
First name	Jon
Middle name	
Last name	Snow
Contact channel	
Contact frequency	

Email

Work: jon.snow@nightswatch.com

Compare with Version 1

CUSTOMER DATA hanges, marked in red.

Salutation	Mr.
First name	John
Middle name	
Last name	Snow

Contact channel Contact frequency Email

john.snow@nightswatch.com





First name	Jon	
	CHANGES	BET
Contact channel Contact frequency		

Email

Work: jon.snow@nightswatch.com



2 changes, marked in red.

Salutation Mr. **First name** John WEEN VERSIONS ----



john.snow@nightswatch.com





First name	Jon
	CHANGES BET
Contact channel Contact frequency Email	Work:

jon.snow@nightswatch.com



2 changes, marked in red.

Salutation Mr. **First name** John WEEN VERSIONS ~~~ WRONG Email

john.snow@nightsatch.com

NO EMAIL BECAUSE INITIAL Email address wrong!

AUDILUGS: HAPPY USERS & LOWER SUPPORT COST

HOW DO WE DO AUDIT LOGS IN JAVA?

FRONT-END **SHOW ALL VERSIONS**

SHOW ONE VERSION

COMPARE 2 VERSIONS

JAVA BACK-END **CREATE VERSION**

QUERY VERSIONS

COMPARE 2 VERSIONS











CODE AUDIT LOGS

JAVERS

Open-source project with Apache license One main contributor

Donations welcome!

Mature with frequent releases (1.0 nearly six years ago)





Computes diff of 2 Plain Old Java Objects (POJOs) Saves POJO versions as JSON in DB (relational DB or MongoDB)

Query versions: POJOs with version data

JAVA BACK-END **CREATE VERSION**

QUERY VERSIONS

COMPARE 2 VERSIONS









CREATE VERSION WHERE DOES IT FIT IN? ADD JAVERS TO PROJECT **UPDATE BACK-END** SAVE VERSION

WHERE DOES TETIN? **DTO:** Data Transfer Object Means: Whatever application sends to UI POJO with fields & ID here Property-value map List of values



BUSINESS LAYER

BUSINESS LAYER

BUSINESS LAYER





BUSINESS LAYER





Validate DTO

BUSINESS LAYER





Validate DTO

Convert for Save

BUSINESS LAYER





Validate DTO

Convert for Save

BUSINESS LAYER

Save

PERSISTENCE LAYER



Validate DTO

Convert for Save

BUSINESS LAYER

PERSISTENCE LAYER

Save

App Tables



Validate DTO

Convert for Save

BUSINESS LAYER

PERSISTENCE LAYER

Save

App Tables





PERSISTENCE LAYER

Save

App Tables






JaVers Tables

Save

PERSISTENCE LAYER

App Tables

DATA STORE



Receive DTO Create Version MY ADVICE: VERSION **EXISTING DTOS**

BUSINESS LAYER

JaVers Tables

PERSISTENCE LAYER



Receive DTO Create Version MY ADVICE: VERSION **EXISTING DTOS**

BUSINESS LAYER

JaVers Tables

BULT- N- BYPASS PERSISTENCE

PERSISIENLE LAYER



WHY VERSION DTOS? Choices for creating version in back-end In persistence transaction, or in new one? In business layer call, or asynchronously?



- Reuses UI: "version DTO" for UI same as existing DTO

 - In persistence data store, or in different one?
- Easy to save as JSON: DTOs already serialized for UI

WHERE DOES TET N?

Version existing DTOs

JaVers bypasses application persistence layer



CREATE VERSION

WIEDE DOES TELLS

ADD JAVERS TO PROJECT

UPDATE BACK-END

SAVE VERSION

ADD JAVERS TO PROJECT Spring Boot Starter for relational DB or MongoDB Integration with Spring JAR in other Java projects Default configuration worked for me



CREATE VERSION

WIEDEDGES TETAS

ADD AVED TO DO LET

UPDATE BACK-END

SAVE VERSION

UPDATE BACK-END

Version storage: Main data source in Spring Boot by default, could be different data store POJO - JSON mapping: Good defaults, but at least 1 annotation/JaVers registration call needed POJO version identity: "Entity ID" (unique per POJO) +

class name





UPDATE BACK-END: EXISTING DTO

public class Customer {

private Long id; private String firstName; private String lastName;

UPDATE BACK-END: UPDATED DTO

public class Customer { 0 Id private Long id; private String firstName; private String lastName;

UPDATE BACK-END **POJO version identity: Unique entity ID per POJO** POJO - JSON mapping: At least @Id for entity ID



CREATE VERSION

WIEDEDGES TETIS

CETTING INVERS

UDDATE BACK END

SAVE VERSION

SAVE VERSION

What: POJO (with entity ID) and user name

How

In persistence transaction, or in new one? In business layer call, or asynchronously?

SAVE VERSION

private Javers javers;

[...]

Customer customer =
 new Customer(50L, "Karsten", "Silz");

this.javers.commit("joecool", customer);

SAVE VERSION

What: POJO (with entity ID) and user name

How: Choices for transaction and synchronicity

D) and user name ion and synchronicity

CREATE VERSION

WIEDEDOES TETNS

CETTIC AVEDS

UDDATE BACK END

SAVE VERSION

JAVA BACK-END

PDEATE VEDEINI

QUERY VERSIONS

COMPARE 2 VERSIONS





QUERY VERSIONS **QUERY ALL VERSIONS QUERY ONE VERSION QUERY NESTED POJOS**

QUERY ALL VERSIONS

Why: List of all versions in UI (version, author timestamp)

Input: JQL (JaVers Query Language) Entity ID and POJO class name Shallow scope": no nested POJOs



QUERY ALL VERSIONS Output: List of "snapshots" Version number ("commit ID"), author, timestamp Map of POJO property values





QUERY ALL VERSIONS: INPUT

JqlQuery query = QueryBuilder
.byInstanceId(50L, Customer.class)
.withShadowScope(ShadowScope.SHALLOW)
.build();

List<CdoSnapshot> versions =
 this.javers.findSnapshots(query);

QUERY ALL VERSIONS: OUTPUT

for (CdoSnapshot aVersion : versions) {

String versionNumber =
 aVersion.getCommitId().value();

CommitMetadata commitMetadata =
 aVersion.getCommitMetadata();
String author = commitMetadata.getAuthor();
LocalDateTime time =
 commitMetadata.getCommitDate();

QUERY ALL VERSIONS Why: List of all versions in UI Input: Entity ID and POJO class name Output: Version number, author, timestamp



QUERY VERSIONS

<u>OLEDY ALL VERSIONS</u>

QUERY ONE VERSION QUERY NESTED POJOS

QUERY ON EVERSION Why: Complete POJO for UI details Input: Same JQL Same Entity ID and POJO class name Version number: From "Query All Versions" Deep scope": All nested POJOs



QUERY ON EVERSION Output: List of "shadows" with 1 element Version number, author, timestamp POJO



QUERY ONE VERSION: INPUT

CommitId commitId = CommitId.valueOf("1.0");

JqlQuery query = QueryBuilder .byInstanceId(50L, Customer.class) .withCommitId(commitId) .withShadowScope(ShadowScope.DEEP_PLUS) .build();

List<Shadow<Customer>> oneVersion =
 this.javers.findShadows(query);

QUERY ONE VERSION: OUTPUT

String versionNumber = aVersion.getCommitId().value();

CommitMetadata commitMetadata = aVersion.getCommitMetadata(); LocalDateTime time = commitMetadata.getCommitDate();

Customer customer = aVersion.get();

- for (Shadow<Customer> aVersion : oneVersion) {
 - String author = commitMetadata.getAuthor();



QUERY ONE VERSION Why: Complete POJO for UI details Input Entity ID and POJO class name Version number Output: POJO

QUERY VERSIONS

<u>CLEDV ONE VERGION</u>

QUERY NESTED POJOS

QUERY NESTED POJOS POJOs can contain other POJOs JaVers stores other POJOs with entity ID separately from parent POJO Impact on queries?



QUERY NESTED POJOS: NEW CLASS

public class Address { **@Id** private Long id;

...

public class Customer { @Idprivate Long id; private Address address;

QUERY NESTED POJOS: SAVE VERSION

CUSTOMER

Version 1

ADDRESS

Version 1

QUERY NESTED POJOS: SAVE VERSION

CUSTOMER

Version 1

ADDRESS

Version 1

CUSTOMER

Update address

ADDRESS Version 2
QUERY NESTED POJOS: SAVE VERSION

CUSTOMER

Version 1

ADDRESS

Version 1



NESTED POJO VERSIONS DON'T UPDATE PARENT POJO VERSION - BUT USERS EXPECT IT!

Update address

CUSTOMER

Version 1 **ADDRESS**

Version 2



QUERY NESTED POJOS

It's a feature, not a bug!

Same in other libraries (e.g., Hibernate Envers)

My Solution

Always change parent POJO to force new version Add "flip-flopping" boolean field (not sent to UI)





QUERY NESTED POJOS

Problem: Nested POJO versions don't update parent POJO version, but users expect it My Solution: Always change parent POJO to force new version





QUERY VERSIONS

<u>COERVONEVERCON</u>

<u>AUERVIERTED DOIOS</u>

JAVA BACK-END

CDEATE VEDEINI Contraction and a start and the second se



COMPARE 2 VERSIONS







COMPARE 2 VERSIONS nput: 2 POJOS Output: List of objects, derived from Change class ValueChange: Simple property change NewObject: Nested POJO added ObjectRemoved: Nested POJO removed Change Count: Sometimes customized per POJO



5) Version Details





Version 3 by Karsten

Sat, 26 September 2020, 14:38

Salutation
First name
Middle name

Last name

Mr. Jon

Snow

Contact channel Contact frequency Email

Work: jon.snow@nightswatch.com

Compare

Version 1



 \sim



Version 1 by Joe

Thu, 17 September 2020, 22:14

2 changes, marked in red.

Salutation	
First name	
Middle name	
Last name	

Contact channel Contact frequency Email

Mr. John

Snow

john.snow@nightswatch.com



DVersion Details





Version 3 by Karsten Sat, 26 September 2020, 14:38

- Last name
- Contact channel Contact frequency Email

Mr. Jon

Snow

Work: jon.snow@nightswatch.com Compare wi

ञ्त

Version 1

USER: 2 CHANGES

2 changes, marked in red.

Salutation First name Middle name

Contact channel Contact frequency Email Mr. John

Snow

<u>john.snow@nights</u> <u>watch.com</u>

Dersion Details



JAVERS: 1 CHANGE



First name

- Middle name
- Last name

Jon

SHOW

- **Contact channel**
- **Contact frequency**

Email

Work: jon.snow@nightswatch.com Compare with

ञ्त

Version 1

USER: 2 CHANGES

2 changes, marked in red.

First name	John
Email	john.snow@ni

john.snow@nightswatch.com



DVersion Details

4

JAVERS: 1 CHANGE

Salutation

First name

Jon

Milia VERS: 2 CHANGES Conta LABEL & EMAL Contact frequency

Email

Work: jon.snow@nightswatch.com Compare with

ð

Version 1

USER: 2 CHANGES

2 changes, marked in red.

First name	John
Email	john.snow@ni

john.snow@nightswatch.com



DVersion Details

JAVERS: 1 CHANGE

First name

JAVERS: 2 CHANGES LABEL & EMAIL

Email

Work: jon.snow@nightswatch.com Compare with

ञ्त

Version 1

USER: 2 CHANGES

2 changes, marked in red.

TURN 3 CHANGES IN JAVERS INTO 2!



john.snow@nights watch.com

.

COMPARE 2 VERSIONS: INPUT

Customer one =
 new Customer(50L, "Karsten", "Silz");
Customer two =
 new Customer(50L, "Joe", "Cool");

Diff differences =
 this.javers.compare(oldOne, newOne);

COMPARE 2 VERSIONS: OUTPUT

Changes listOfDifferences =
 differences.getChanges();

for (Change aChange : listOfDifferences) {
 System.out.println(aChange.toString());
}

COMPARE 2 VERSIONS: CONSOLE

ValueChange{ 'firstName' value changed from
'Karsten' to 'Joe' }

ValueChange{ 'lastName' value changed from
'Silz' to 'Cool' }

COMPARE 2 VERSIONS

- Input: 2 POJOS
- Output: List of Change instances
- Change Count: Sometimes customized per POJO



JAVA BACK-END

PDEATE VEDEINN A Standard Standard and a Read State and a stand a stand a state and a state a state a state a state a state a

NIEDV VEDEINNE

COMPADE 2 VEDSIONS









SHOWALLYERSONS

New UI, generic with version information

SHOW ONE VERSION

COMPARE 2 VERSIONS

JAVA BACK-EN

CONDADE 2 VEDENSE Societations and a second second second





SHOWALLY ENDER STORES

New UI, generic with version information

SHOW OF A CALL OF A CALL

COMPARE 2 VERSIONS

JAVA BACK-EN

CONDADE 2 VEDENSE Societations and a second second second





SHOWALLY BRANCHER STORES

New UI, generic with version information

SHOW OF A CONTRACT OF A CONTR

COMPARE 2 VEDSIONS Extends UI with diff

JAVA BACK-EN

CONDADE 2 VEDENSE Societations and a second second second





Statute States S

version information

Show Stranger Show Reuses UI, version information

Extends Ul with diff

JAVA BACK-EN

CONDADE 2 VEDCON





AUDIT LOG: WHO CHANGED WHAT WHEN HOW

AUDIT LOGS: HAPPY USERS & LOWER SUPPORT COST

JAVERS CREATES, STORES & QUERIES VERSIONS AND COMPUTES DIFFS – WE HAVE LESS CODE TO WRITE

VERSION EXISTING DTOS – WE REUSE CODE

5 **Version Details**





Version 3 by Karsten

Sat, 26 September 2020, 14:38

- Salutation First name
- Middle name
- Last name

Mr. Jon



Contact channel Contact frequency Email

Work: jon.snow@nightswatch.com

Compare



Version 1



 \sim

 \leftarrow

Version 1 by Joe Thu, 17 September 2020, 22:14 2 changes, marked in red. ition name **Inddle name** Last name

Mr. John

Snow

Contact channel Contact frequency Email

john.snow@nightswatch.com



CODE AUDIT LOGS

WHAT'S IN THE SAMPLE CODE?

More on Spring Boot configuration

More POJO mapping customizations (ignoring or renaming fields)

How to reduce POJO JSON size in data storage

What JaVers writes to its tables





SAMPLE CODE B APPS TOD

