

## EDIT - feature request #9862

### centralized password policy enforcement validator

11/12/2021 02:37 PM - Andreas Kohlbecker

<b>Status:</b>	Closed	<b>Start date:</b>	
<b>Priority:</b>	New	<b>Due date:</b>	
<b>Assignee:</b>	Andreas Kohlbecker	<b>% Done:</b>	100%
<b>Category:</b>	cdmlib	<b>Estimated time:</b>	0:00 hour
<b>Target version:</b>	Release 5.29		
<b>Severity:</b>	normal		
<b>Description</b>			
Currently rules to validate the security of passwords are only implemented in the TaxEditor (eu.etaxonomy.taxeditor.ui.password.PasswordWizardPage.PasswordValidator).			
For consistent password security rules a class to validate the password security should be implemented centrally in the cdmlib.			
<a href="#">Passay</a> is an often used password policy enforcement library for java			
<b>Related issues:</b>			
Related to EDIT - feature request #9859: Password Recovery UI		Closed	

#### Associated revisions

##### Revision 4b4dff4 - 11/12/2021 03:58 PM - Andreas Kohlbecker

fix #9862 implementing central password constraint validation classes and annotation

##### Revision 096c0a8e - 11/12/2021 04:03 PM - Andreas Kohlbecker

fix #9862 fixing imports

##### Revision ef2b56b4 - 11/15/2021 03:49 PM - Andreas Kohlbecker

ref #9862 better password constraint violation message concatenation

##### Revision a8e937e5 - 01/21/2022 03:28 PM - Andreas Müller

ref #9862 rename password validator class, add tests, add level2 validation to User, add maxLength and handle null

##### Revision 1ed85787 - 01/21/2022 03:43 PM - Andreas Müller

ref #9862 add javadoc link to ticket

##### Revision 20cedc11 - 01/21/2022 03:49 PM - Andreas Müller

ref #9862 adapt to new name for password validator

##### Revision c603d6b3 - 01/26/2022 11:59 AM - Andreas Müller

ref #9862 fix NPE in ValidPasswordValidator

#### History

##### #1 - 11/12/2021 02:38 PM - Andreas Kohlbecker

- Related to feature request #9859: Password Recovery UI added

##### #2 - 11/12/2021 02:42 PM - Andreas Kohlbecker

- Description updated

##### #3 - 11/12/2021 03:58 PM - Andreas Kohlbecker

- Status changed from New to Resolved

- % Done changed from 0 to 50

Applied in changeset [cdmlib|4b4dff4929970b5b547c1f21b10359fd3f39c50](#).

**#4 - 11/12/2021 04:05 PM - Andreas Kohlbecker**

the [4b4dfff4](#) also contains a @ValidPassword annotation which can be used at model classes

please review

**#5 - 11/12/2021 04:05 PM - Andreas Kohlbecker**

- Assignee changed from Andreas Kohlbecker to Andreas Müller

**#6 - 01/21/2022 02:14 PM - Andreas Müller**

- Description updated

**#7 - 01/21/2022 03:36 PM - Andreas Müller**

- Status changed from Resolved to Feedback

- Assignee changed from Andreas Müller to Andreas Kohlbecker

- % Done changed from 50 to 80

Generally it works. Some minor issues:

- The validator class should always named like {ValidationAnnotationName}Validator, I did rename it to ValidPasswordValidator
- I added null check as PasswordData throws exception for null value
- I added max length as the password field in the DB only accepts 256 characters (and also the error message does not look nice if saying '... should be between 8 and 210000000 or so)
- tests were still missing, I added some first tests

Also validation did not yet use hibernate bean validation at all yet. This is necessary for Level1 validation as existing data may create errors otherwise. But we can use Level2 validation already and in future switch to Level1 validation. I added the according annotation to User.password field.

Note: Generally vaadin still seems to workaround the integrated hibernate bean validation by using the public static method in the validator. In future we may want to integrate bean validation deeper into vaadin UIs. For now this is perfectly ok as it allows fast implementation but code is still at the correct place in model.validation.

Please review if you agree with my minor changes.

**#8 - 01/21/2022 03:47 PM - Andreas Müller**

One more issue. I wonder if we really need to integrate a full new library for this function. There seem to be simple regex solutions which are more lightweight like here: <https://stackoverflow.com/questions/10877668/password-validation-using-hibernate-bean-validation>

**#9 - 01/21/2022 04:51 PM - Andreas Kohlbecker**

- File clipboard-202201211646-6i1sw.png added

- File clipboard-202201211648-goxaw.png added

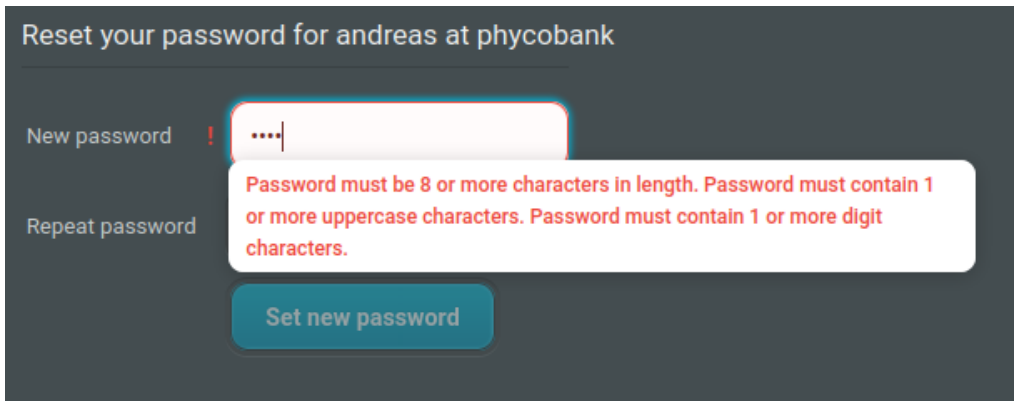
- Status changed from Feedback to Resolved

Andreas Müller wrote in [#note-8](#):

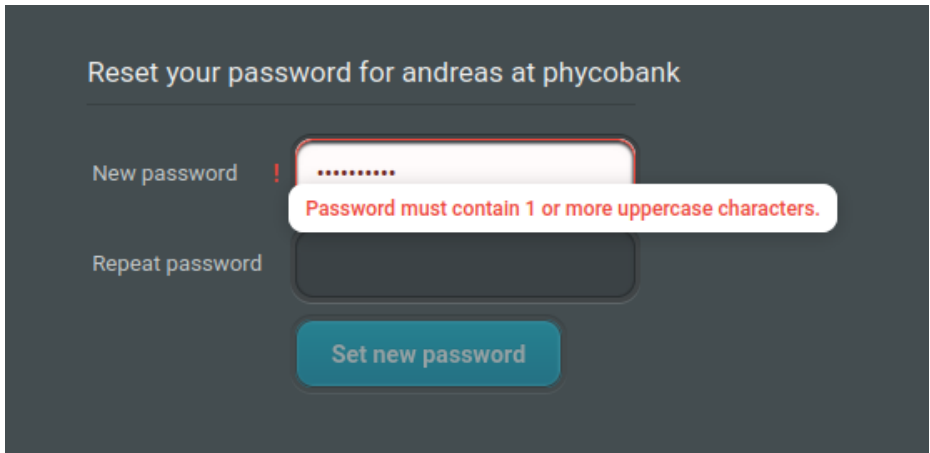
One more issue. I wonder if we really need to integrate a full new library for this function. There seem to be simple regex solutions which are more lightweight like here: <https://stackoverflow.com/questions/10877668/password-validation-using-hibernate-bean-validation>

Password rules can be fairly complex, so that it is not always easily understandable why a password has been rejected. This library helps a lot to identify what is wrong:

password = "aaa"



password = "1aaaaaaa"



"1aaaaaaaaA" is still not a good password, so it would make sense to add rules for variation, or entropy. This is also supported by this library and nothing you can achieve with regexes.

BTW: I will do the review of your implementation next week.

**#10 - 01/21/2022 11:42 PM - Andreas Müller**

Yes I agree, it is probably better although for possible future developments to have a more sophisticated library at hand.

**#11 - 01/27/2022 03:42 PM - Andreas Kohlbecker**

- Status changed from Resolved to Closed
- % Done changed from 80 to 100

I agree with your modifications and all open questions are solved, so we can close this ticket now

**#12 - 02/15/2022 03:40 PM - Andreas Müller**

- Target version changed from Release 5.45 to Release 5.29

**Files**

clipboard-202201211646-6i1sw.png	28.9 KB	01/21/2022	Andreas Kohlbecker
clipboard-202201211648-goxaw.png	22.4 KB	01/21/2022	Andreas Kohlbecker