Validator

notEmpty

notEmpty(‘name’, $message = null,$when =null)

$validator->notEmpty('title', 'Title is not empty')

allowEmpty

->allowEmpty($field, $when = null, $message == null)

requirePresence ~ check isset()

$validator->requirePresence(‘name’);

Length – string, number

minLength

$validator->add('title', 'length', [

'rule' => ['minLength', 8],

'message' => \_\_('Title must 8 character')

])

maxLength

$validator->add('title', 'length', [

'rule' => ['maxLength', 8],

'message' => 'Title must 8 character'

])

lengthBetween

Custom validation rule ~ CallBack

* Read more: https://book.cakephp.org/3.0/en/core-libraries/validation.html#custom-validation-rules

// check password is same repassword

//context chứa all input: $context[‘data’][….inputName….]

$validator->add('password', 'custom', [

'rule' => function($value, $context) {

if($value == $context['data']['repassword']) {

return true;

}

return false;

},

'message' => \_\_('Password is not same Repassword')

]);

Unique check form Database

* Syntax:

Validateunique($value, array $options, array $context = [])

* Read more: https://api.cakephp.org/3.0/class-Cake.ORM.Table.html#\_validateUnique
* Parameters:
* $value: The value of column to be checked for uniqueness
* Array $options: The options array, optionally containing the 'scope' key. May also be the validation context if there are no options
* Array $context: Either the validation context or null
* Example
* unique email

$validator->add('email', [

'unique' => ['rule' => 'validateUnique', 'provider' => 'table']

])

* Unique validation can be scoped to the value of another column

$validator->add('email', [

'unique' => [

'rule' => ['validateUnique', ['scope' => 'site\_id']],

'provider' => 'table'

]

]);

* Note: don’t use in Form class, if want using then create custom rule, where select and check normal.

// example for RegisterForm

//create user table

protected $Users;

public function \_\_construct()

{

$this->Users = TableRegistry::get('Users');

}

//

$validator

->add('email', [

'uniqueEmail' => [

'rule' => function($value, $context){

return $this->uniqueEmail($value);

},

'message' => \_\_('Email not isvalid')

],

]);

// finish