'username' => Yii::t('app/user', 'Username'), 'password' => Yii::t('app/user', 'Password'), 'rememberMe' => Yii::t('app/user', 'Remember Me'),

];

}

}

Also, change the attribute labels of the LoginForm model:

class ContactForm extends Model {

public function attributeLabels()

{

return [

'name' => Yii::t('app/contact', 'Name'),

'email' => Yii::t('app/contact', 'Email'),

'subject' => Yii::t('app/contact', 'Subject'),

'body' => Yii::t('app/contact', 'Body'),

'verifyCode' => Yii::t('app', 'Verification Code'),

];

}

}

It will output translated labels for the current language instead of originals.

5. To prepare translations, create the messages directory. Right now, we can create translation files for all needed languages. We can do it manually, but there is a helpful crawler that can scan all project files and extract all messages from Yii ::t() constructions. Let’s use it.

6. Generate the configuration file for the message scanner:

./yii message/config-template config/messages.php

7. Open the configuration file and set the following values:

<?php return [

'sourcePath' => '@app',

'languages' => ['de', 'fr'],

'translator' => 'Yii::t',

'sort' => false,

'removeUnused' => false,

'markUnused' => true,

'only' => ['\*.php'],

'except' => [

'.svn',

'■git',

'■gitignore',

'.gitkeep',

'■hgignore',

' . hgkeep',

'/messages',

'/vendor',

],

'format' => 'php',

'messagePath' => '@app/messages',

'overwrite' => true, 'ignoreCategories' => [

'yii',