'class' => 'yii\caching\FileCache',

],

],

You can use any cache storage instead of FileCache. Also, you can register multiple cache application components and use Yii: :$app->cache and Yii: :$app->cache2 for different data types:

'components' => [

'cache' => [

'class' => 'yii\caching\MemCache',

'useMemcached' => true,

],

'cache2' => [

'class' => 'yii\caching\FileCache',

],

],

The framework uses the cache component by default in its own classes.

1. Enable table schema caching for the db component as follows:

return [

// ...

'components' => [

// ...

'cache' => [

'class' => 'yii\caching\FileCache',

],

'db' => [

'class' => 'yii\db\Connection',

'dsn' => 'mysql:host=localhost;dbname=mydatabase',

'username' => 'root',

'password' => '',

'enableSchemaCache' => true,

// Optional. Default value is 3600 seconds schemaCacheDuration' => 3600,

// Optional. Default value is 'cache'

'schemaCache' => 'cache',

],

],

];

1. Use plain arrays instead of Active Record objects for listing sets of elements:

$categoriesArray = Categories::find()->asArray()->all();

1. Use each () instead of all() in foreach for a large count of results:

foreach (Post::find()->each() as $post) {

// ...

}

1. Because Composer’s autoloader is used to include most third-party class files, you should consider optimizing it by executing the following command:

composer dump-autoload -o