When using a memcached backend, you should take into account the fact that when using these solutions the application user can possibly lose the session if the maximum cache capacity is reached.

Note

Note that, when using a cache backend for a session, you cannot rely on a session as a temporary data storage, since then there will be no memory to store more data in memcached. In such a case, this will just purge all data or delete some of it.

If you are using multiple servers, you cannot use file storage. There is no way to share the session data between servers. In the case of memcached, it is easy because it can be easily accessed from as many servers as you want.

Also, for sharing the session data you can use DbSession:

return [

// ...

'components' => [

'session' => [

'class' => 'yii\web\DbSession',

],

],

];

Now, create a new table in your database:

CREATE TABLE session (

id CHAR(40) NOT NULL PRIMARY KEY, expire INTEGER, data BLOB

)

There’s more...

It is a good idea to close the session as soon as possible. If you’re not going to store anything in the session during the current request, you can even close it at the very beginning of your controller action. This way, even when using files as storage your application should be fine.

Use the following command:

Yii:$app->session->close();

See also

For more information about performance and caching refer to the following URLs:

* [http://www.yiiframework.com/doc-2.0/guide-tutorial-performance-tnning.html](http://www.yiiframework.com/doc-2.0/guide-tutorial-performance-tuning.html)
* <http://www.yiiframework.com/doc-2.0/guide-caching-overview.html>