type: ' post', dataType: ' j son',

data: {paraml: paraml, \_csrf : csrfToken},

Additionally [rename]

});

If your application requires a very high security level, such as a bank account management system, extra measures can be taken.

First, you can turn off the remember me feature using config/main .php, as follows:

'components' => [

'user' => [

'enableAutoLogin' => false,

].

],

Note that this will not work if the enabledSession option is true.

Then, you can lower the session timeout, as follows:

'components' => [

'session' => [

'timeout' => 200,

],..

],

This sets the number of seconds after which data will be seen as garbage and cleaned up.

Of course, these measures will make the user experience worse, but they will add an additional level of security.

Using GET and POST properly

HTTP insists on not using get operations that change data or state. Sticking to this rule is good practice. It will not prevent all types of CSRF, but it will at least implement some injections, such as <img src=, pointless>.

See also

In order to learn more about SQL injections and working with databases through Yii, refer to the following URLs:

[http://en.wikipedia.org/wiki/Cross-site request forgery](http://en.wikipedia.org/wiki/Cross-site_request_forgery)

[http://www.viiframework.com/doc-2.0/guide-securitv-best-practices.html#avoiding-csrf](http://www.yiiframework.com/doc-2.0/guide-security-best-practices.html%23avoiding-csrf)