As an alternative or an addition to the Last-Modified header variable, you can use ETag.

**Entity Tag**

In cases when we do not store the last modified date in our documents or pages, we can use custom hashes, which can be generated at the base of the document content.

For example, we can use a content title for our document to hash a specific tag:

class BlogController extends Controller {

public function behaviors()

{

return [

[

'class' => 'yii\filters\HttpCache',

'only' => ['view'],

'etagSeed' => function ($action, $params) {

$article = $this->findModel(\Yii::$app->request->get('id')); return serialize([$article->title, $article->text]);

},

],

];

}

// ...

}

The HttpCache filter will attach this tag to the server response as an ETag header variable.

After storing ETag, our browser can attach it in the If-None-Match header for every subsequent request.

If the document still has not changed, HttpCache will send a lightweight 304 response header without running the action.

|  |  |
| --- | --- |
| Request | Response |
| First request with full response | |
| GET index.php?r=blog/view&id=3 HTTP 1.1 | HTTP/1.1 200 OK  Cache-Control: public, max-age=3600 Etag: "VYkwdOXBzV23KhnzTTJXU"  <!DOCTYPE html>  <html lang="en-US"> |
| Second request with If-None-Match and blank response | |
| GET index.php?r=blog/view&id=3 HTTP 1.1 | HTTP/1.1 304 Not Modified |