

Pages / ... / yii2相关

yii2-rateLimiter

Created by 杨超, last modified on 2018 Oct 12

- 概述
- 使用
 - edc中的使用
 - 1 首先实现了 RateLimiterUser
 - 2 重载一下原始的RateLimiter
 - 3 在controller里使用示例
 - 后续扩展

概述

参考yii2使用文档: <https://www.yiichina.com/doc/guide/2.0/rest-rate-limiting>

简单来说 可以对api进行速度限制, 原理是 基于 滑动时间窗口的计数来进行频率的限制, 如果超限将返回响应状态代码 429

使用

理论上只要有一个实现了接口 `yii\filters\RateLimitInterface` 的User, 可以支持获取 流量上限 以及 【获取 / 保存】当前的流量计数就可以了

```
public function getRateLimit($request, $action)
{
    return [$this->rateLimit, 1]; // $rateLimit requests per second
}

public function loadAllowance($request, $action)
{
    return [$this->allowance, $this->allowance_updated_at];
}

public function saveAllowance($request, $action, $allowance, $timestamp)
{
    $this->allowance = $allowance;
    $this->allowance_updated_at = $timestamp;
}
```

```
$this->save();  
}
```

edc中的使用

1 首先实现了 RateLimiterUser

来实现 `yii\filters\RateLimitInterface`

没啥好提的，就是使用业务key + app-controller-action 做key，基于redis save/get 当前的流量值

2 重载一下原始的RateLimiter

方便配置不同的业务方，也为未来的可配置化做准备

```
class RateLimiter extends \yii\filters\RateLimiter  
{  
    public $key = '';  
  
    /**  
     * [  
     *      'action' => [  
     *          'key1' => $limit1,  
     *          'key2' => [$limit2, $period],  
     *          ...  
     *      ],  
     *      ...  
     * ]  
     * @var array  
     */  
    public $rateMap = [];  
  
    public function init()  
    {  
        parent::init();  
        $this->errorMessage = '访问频率超限';  
        $this->key = $this->getCurrentKey();  
    }  
}
```

```
$user = new RateLimiterUser([
    'key' => $this->key,
]);
$this->user = $user;
}

public function beforeAction($action)
{
    if (isset($this->rateMap[$action->id])) {
        if ($this->checkKey($this->rateMap[$action->id])) {
            $limit = $this->rateMap[$action->id][$this->key];
            is_int($limit) && $limit = [$limit, 1];
            $this->user->setRateLimit($limit);
            \Yii::$app->response->format = \yii\web\Response::FORMAT_JSON;
            return parent::beforeAction($action);
        }
    }
    return true;
}

/**
 * 获取当前业务方的key
 * @return string
 */
public function getCurrentKey()
{
    if (\Yii::$app->request->isGet) {
        parse_str(\Yii::$app->request->queryString, $params);
    } else {
        $params = \Yii::$app->request->post();
    }
    return ArrayHelper::getRightValue($params, '_key', '');
}

/**
```

```
* 判断是否是要判断的业务方key 并且 key限制速率不为0
* @return bool
*/
public function checkKey($arr)
{
    return isset($arr[$this->key]) && ($arr[$this->key] != 0);
}
```

3 在controller里使用示例

说明一下 业务方是abcfintech 【阿博茨公司】

```
public function behaviors() {
    ...
    $rateMaps = [
        'out-list' => ['abcfintech' => [100, 60]],
        'out-detail' => ['abcfintech' => [100, 60]],
        'out-delete-list' => ['abcfintech' => [100, 60]],
    ];
    return [
        'edcOutAbcRateLimit' => [
            'class' => RateLimiter::class,
            'rateMap' => $rateMaps,
        ]
    ];
    ...
}
```

后续扩展

可预期的未来，随着edc数据质量和覆盖度的提升，很可能会接入更多的业务方，到时候需要将目前手写代码控制的配置 使用db管理起来，做一个可配置化的平台

Like Be the first to like this

No labels

地址：北京市朝阳区建国路86号佳兆业广场北塔6层梦想加空间601室

以太资本由艾普拉斯投资顾问(北京)有限公司运营，提供早期互联网项目的投融资对接服务

©2014-2017 以太资本 京ICP备14028208号