ddp-rate-limiter

Source code of released version | Source code of development version ***

DDP Rate Limiter package

A rate limiter added directly to DDP that provides an API to add rules to Meteor methods and collections.

For example usage, check out the docs.

Pre-defined Defaults

If the `accounts-base` package is added to your project, there are default rules added to limit logins, new user registration and password resets calls to a limit of 5 requests per 10 seconds per connection. These provide a basic solution to dictionary attacks where a malicious user attempts to guess the passwords of legitimate users by attempting all possible passwords. To remove the default rule, a user can add `Accounts.removeDefaultRateLimit()` to any server side code and the default rate limit will be removed.

Configuration

The `DDPRateLimiter` is configured with a set of rules. Each rule is a set of keys to be inspected with filters on those keys to specify all DDP messages that satisfy the rule. Each of these possible messages that satisfy the rule is given a bucket by creating a unique string composed of all the keys in the rule and the values from the message. After each rule's specified time interval, all the buckets are deleted. A rate limit is said to have been hit when a bucket has reached the rule's capacity, at which point errors will be returned for that input until the buckets are reset.

A rule is defined as a set of key-value pairs where the keys are one or more of `userId`, `clientAddress`, `type`, `name`, and `connectionId`. The values can either be null, primitives or functions. When you want to rate limit some users but not others, a rule can match invocations using a function in a way that is determined at run time based on the database or other data. In our example, we check the database to avoid rate limiting admin users.

When we add the rule to DDPRateLimiter, we also specify the number of messages that we allow and the time interval on which the rate limit is reset.