```
worker_processes  2;
error_log logs/error.log warn;

events {
    worker_connections 1024;
}

http {
    upstream backend {
        drizzle_server 127.0.0.1:3306 protocol=mysql
                       dbname=ngx_test user=ngx_test password=ngx_test;
        drizzle_keepalive max=10 overflow=ignore mode=single;
    }

    server {
        listen 8080;

        location @cats-by-name {
            set_unescape_uri $name $arg_name;
            set_quote_sql_str $name;
            drizzle_query 'select * from cats where name=$name';
            drizzle_pass backend;
            rds_json on;
        }

        location @cats-by-id {
            set_quote_sql_str $id $arg_id;
            drizzle_query 'select * from cats where id=$id';
            drizzle_pass backend;
            rds_json on;
        }

        location = /cats {
            access_by_lua '
                if ngx.var.arg_name then
                    return ngx.exec("@cats-by-name")
                end

                if ngx.var.arg_id then
                    return ngx.exec("@cats-by-id")
                end
            ';
            rds_json_ret 400 "expecting \"name\" or \"id\" query argume
        }
    }
}
```

And then we start our Nginx server with this configure file and test with our`/cats` service like this:

```
$ curl 'localhost:8080/cats'
{"errcode":400,"errstr":"expecting \"name\" or \"id\" query arguments"}

$ curl 'localhost:8080/cats?name=bob'
[{"id":3,"name":"bob"}]

$ curl 'localhost:8080/cats?id=2'
[{"id":2,"name":null}]
```

The actual output rows may vary depending on the actual contents in your `cats` table anyway.