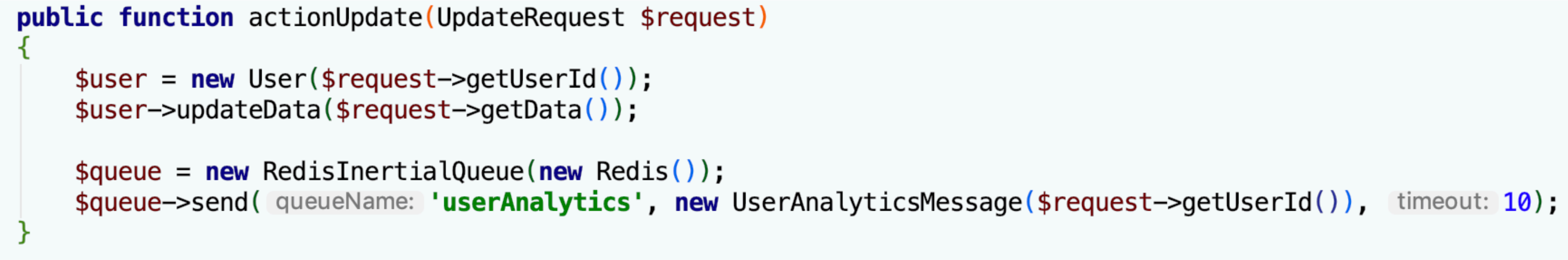
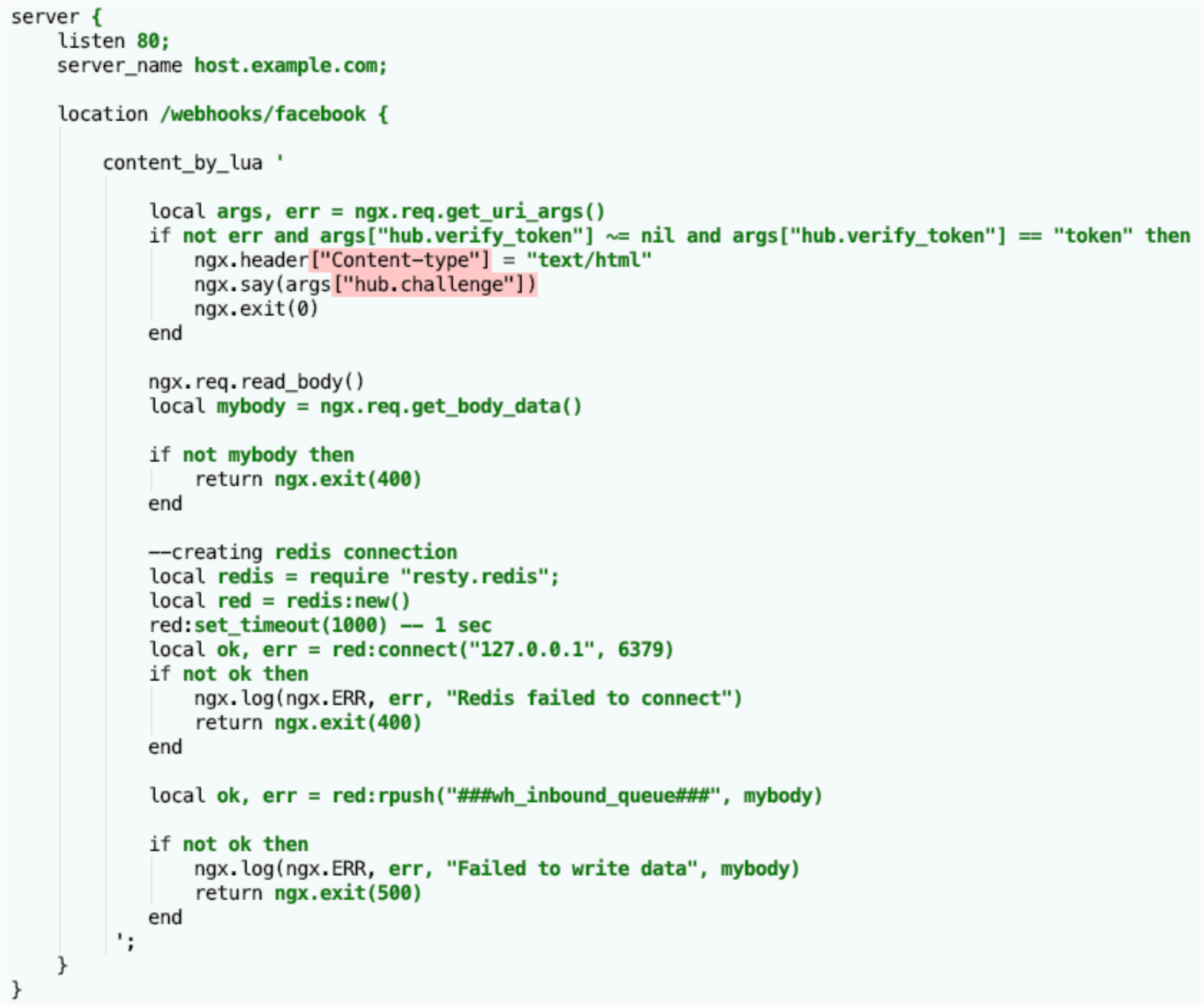
Инерциальные очереди



```php
public function actionUpdate(UpdateRequest $request)
{
    $user = new User($request->getUserId());
    $user->updateData($request->getData());

    $queue = new RedisInertialQueue(new Redis());
    $queue->send( queueName: 'userAnalytics', new UserAnalyticsMessage($request->getUserId()),  timeout: 10);
}
```

Очереди из NGINX



```
server {
    listen 80;
    server_name host.example.com;

    location /webhooks/facebook {

        content_by_lua '

            local args, err = ngx.req.get_uri_args()
            if not err and args["hub.verify_token"] ~= nil and args["hub.verify_token"] == "token" then
                ngx.header["Content-type"] = "text/html"
                ngx.say(args["hub.challenge"])
                ngx.exit(0)
            end

            ngx.req.read_body()
            local mybody = ngx.req.get_body_data()

            if not mybody then
                return ngx.exit(400)
            end

            --creating redis connection
            local redis = require "resty.redis";
            local red = redis:new()
            red:set_timeout(1000) -- 1 sec
            local ok, err = red:connect("127.0.0.1", 6379)
            if not ok then
                ngx.log(ngx.ERR, err, "Redis failed to connect")
                return ngx.exit(400)
            end

            local ok, err = red:rpush("###wh_inbound_queue###", mybody)

            if not ok then
                ngx.log(ngx.ERR, err, "Failed to write data", mybody)
                return ngx.exit(500)
            end
         ';
    }
}
```