

Customer HTTPS Certificates with Ploi

If you haven't set up a Ploi site before, watch this video: [Deploying SaaS boilerplate to Ploi](#)

Ploi added a special feature for us: generating and revoking HTTPS certificates for client domains.

Ploi setup

If you use the Netlify model for client domains, you start by giving users a subdomain on your central domain. Then you let them add their custom domains.

To get the subdomains working, you need to set up:

- An alias for `*.yourcentraldomain.com`, such that the webserver listens to that domain
- An HTTPS certificate for `*.yourcentraldomain.com, yourcentraldomain.com`, such that the content on subdomains will be served using HTTPS. You will need a specific DNS provider (CloudFlare is one) to set this certificate up.

Domains

We can only **generate** and **revoke** certificates. This means that we should only be able to **create** and **delete** domains. No updating.

Make sure domains can't be deleted. For example using Eloquent event listeners.

The [boilerplate](#) has this implemented in a very nice way. Users can change their subdomains, but domains can't be updated. They can only be created and deleted.

Once a domain is created, users can click a button to issue certificate generation. When a certificate is generated, they can click a button to revoke it.

API calls

When creating a domain, you need two calls.

First, add the domain to Ploi.

Second, generate a certificate for that domain.

But you cannot do this at once — you could run into race conditions. Please be 100% sure you do these actions separately. Have different buttons in the UI for them, etc.

The code examples below come from the boilerplate. Properties like `$this->server` are the server id (you can find it in the URL). In case of the boilerplate, they're read from the env (via a config of course).

There will be some specific calls like `$domain->isSubdomain()`. Please add these checks such that they fit your application setup.

Adding a domain

```
/**
 * Add tenant :80 vhost
 *
 * @param Domain $domain
 * @return boolean
 */
public function addDomain(Domain $domain): bool
{
    if ($domain->isSubdomain() || ! $this->token) {
        return false;
    }
}
```

```

    }

    if (gethostbyname($domain→domain) === gethostbyname(Domain::domainFromIp($domain→ip))) {
        return false;
    }

    Http::withToken($this→token)→asJson()
        →post("https://ploi.io/api/servers/{$this→server}/sites/{$this→site}/tenants",
            ['tenants' => [$domain→domain],
            ]
        );

    return true;
}

```

Removing a domain

```

/**
 * Remove a tenant :80 host.
 *
 * @param Domain $domain
 * @return boolean
 */
public function removeDomain(Domain $domain): bool
{
    if ($domain→isSubdomain() || ! $this→token) {
        return false;
    }

    Http::withToken($this→token)→asJson()
        →delete("https://ploi.io/api/servers/{$this→server}/sites/{$this→site}/tenants/{$domain→id}");

    return true;
}

```

Requesting a certificate

```
/**
 * Request a certificate for a tenant host.
 *
 * @param Domain $domain
 * @return boolean
 */
public function requestCertificate(Domain $domain): bool
{
    if ($domain->isSubdomain() || ! $this->token) {
        return false;
    }

    Http::withToken($this->token)->asJson()
        ->post("https://ploi.io/api/servers/{ $this->server }/sites/{ $this->site }/certificates", [
            'webhook' => tenant()->route('tenant.ploi.certificate.issued')
        ])
        ->send();

    $domain->update(['certificate_status' => 'pending']);

    return true;
}
```

Revoking a certificate

```
/**
 * Revoke a certificate for a tenant host.
 *
 * @param Domain $domain
 * @return boolean
 */
public function revokeCertificate(Domain $domain): bool
{

```

```
if ($domain→isSubdomain() || ! $this→token) {  
    return false;  
}  
  
Http::withToken($this→token)→asJson()  
    →post("https://ploi.io/api/servers/{$this→server}/sites/{$this→site}/  
        'webhook' ⇒ tenant()→route('tenant.ploi.certificate.revoked'  
    ]  
);  
  
$domain→update(['certificate_status' ⇒ 'pending']);  
  
return true;  
}
```

DOCUMENTATION

Tenants

Event system

Configuration

BUSINESS

SaaS boilerplate

Consulting

Audits

DOCUMENTATION

Compared to other packages

Integrations

Tenant identification

LINKS

Branding

GitHub

Discord

Donate

SUBSCRIBE TO OUR NEWSLETTER

Receive notifications about important releases, new packages and other updates.

Enter your email

Subscribe



© 2020 Samuel Štancl. All rights reserved.