



PHP class for our service



Chrome extension for our service



Juju-based charms for Ubuntu



Our favorite: Drupal module :-)



What we'll make available for our users.

...



Drupal-services-based REST-ful API for final



Drupal-based access interface



Bash-based prog. for generating PDFs

What we'll deploy on our servers.

In the beginning, we'll stick to MD5. Thus:
 $APIKey = md5(E\text{-mail ID} + Password\ MD5)$

It's very simplistic yet robust.

This is the list of authorized domains. This will be a multi-value field.

In the beginning, we'll have only free. But soon, we'll open up for:

- Free
- Premium (Paid)
- Epic (Paid)

A single content-type to store all membership information.

Drupal user ID	API key	Domains	Membership type
----------------	---------	---------	-----------------

We will create an SHA1 key of the contents of the PDF + params.

1. This will guarantee that it's unique; if it's the same, the document will be the same.

2. We can use this in the URL as it's also cryptographic. For example,
www.hopus.com/docs/6ff87c4664981e4397625791c8ea3bbb5f2279a3

3. This will also help us in developing a custom cache strategy. Will explain later.

Since we are using *WKHTMLTOPDF*, we can allow people to customize the document as per their requirements.

This can take one of the following two values:

1. URI (We are to generate the PDF from a publicly accessible URL)
2. HTML (We are to generate the PDF using the passed HTML)

Our data store.

Drupal user ID	Document SHA Key	Params	Origin domain	Type	Source URI	Document file ID
----------------	------------------	--------	---------------	------	------------	------------------

The IP or the domain from where the document originated.

If $Type == URI$ then this field will be populated.

If $Type == HTML$, we will save the HTML document as a file and keep the file ID in this field.



Bash-based prog. for
generating PDFs

```
pdfgen -type='URI' -params="" -uri='http://www.google.com'
```

```
pdfgen -type='HTML' -params="" -path="\var\www\pdfGen\6ff87c4664981e4397625791c8ea3bbb5f2279a3.html"
```

1. **Why bash:** Will be a fire-and-forget mechanism for us.

2. Since the call will be made only by our Drupal application, it will be made post authentication. So, effectively, this script will be logidless. It's core requirement is speed.

2. Bash has some size limitations. But we can override them at the OS level. See this link
(<http://www.cyberciti.biz/faq/argument-list-too-long-error-solution/>).