

# Assetic - Holly Shit

Redexperts.net

@emgiezet

# TL;DR;

- Manage Assets
  - JavaScripts
  - CSS + Less + Sass
  - Images
- Flow in Dev Mode
  - Wooden PC
  - Normal PC
- Flow in Production
  - CDN Support

# JavaScripts

- Combine
- Get Single file output
- Minify & Apply Filters

The image shows the JavaScript logo, which consists of the letters 'J' and 'S' in a bold, black, sans-serif font. The letters are positioned on a solid yellow square background. The 'J' is on the left and the 'S' is on the right, both centered vertically within the square.

JS

# In your template you got this

```
{% javascripts '@AcmeFooBundle/Resources/public/js/*' %}  
    <script type="text/javascript" src="{{ asset_url }}"></script>  
{% endjavascripts %}
```

1. Get all the files from @AcmeFooBundle/Resources/public/js/\*
2. Apply filters
3. Put output in: <script type="text/javascript" src="{{ asset\_url }}"></script>
4. Badum TSSS

# You Can Combine!

```
{% javascripts
  '@AcmeFooBundle/Resources/public/js/*'
  '@AcmeBarBundle/Resources/public/js/form.js'
  '@AcmeBarBundle/Resources/public/js/calendar.js' %}
<script src="{{ asset_url }}"></script>
{% endjavascripts %}
```

- Multi files in one output == Only 1 HTTP request for your apache

# You can specify your output name!

```
{% javascripts '@AcmeFooBundle/Resources/public/js/*'  
output='js/compiled/main.js' %}  
    <script src="{{ asset_url }}"></script>  
{% endjavascripts %}
```

- All steps before will be compiled into: `js/compiled/main.js`

# Apply Filters

```
assetic:
```

```
  filters:
```

```
    yui_js:
```

```
      jar: "%kernel.root_dir%/Resources/java/yuicompressor.jar"
```

```
{% javascripts '@AcmeFooBundle/Resources/public/js/*' filter='yui_js' %}
```

```
  <script src="{{ asset_url }}"></script>
```

```
{% endjavascripts %}
```

- All files will be minified by yuicompressor

# Coffee? Why not.

```
# app/config/config.yml
assetic:
  filters:
    coffee:
      bin: /usr/bin/coffee
      node: /usr/bin/node
      apply_to: "\.coffee$"
```

# CSS



# Same thing

```
{% stylesheets 'bundles/acme_foo/css/*'  
filter='cssrewrite' %}  
    <link rel="stylesheet" href="{{ asset_url }}" />  
{% endstylesheets %}
```

- Yeah you know the syntax

# Sass? Less? Why not!

```
# app/config/config.yml
parameters:
  # Assetic
  assetic.filter.compass.images_dir:
%kernel.root_dir%/../web/images
  assetic.filter.compass.http_path: /images

assetic:
  debug:          %kernel.debug%
  use_controller: false
  filters:
    sass:      ~
    compass:   ~
```

```
# app/config/config.yml
assetic:
  debug:          %kernel.debug%
  use_controller: false
  filters:
    cssrewrite: ~
    less:
      node: /usr/local/bin/node
      node_paths: [/usr/local/lib/node]
    yui_css:
      jar: %kernel.root_dir
%../java/yuicompressor-2.4.6.jar
    yui_js:
      jar: %kernel.root_dir
%../java/yuicompressor-2.4.6.jar
```

# Images



```
{{ asset('images/cats.jpg') }}
```

**\*\*Badum TSSS\*\***

# Dev Flow

- Refresh all assets per request don't cache them
- Any change in assets code should appear **immediately** on dev page
- Costs:
  - Site is generating a bit slower than on Prod ( about 10 times )

# Dev (moar than 2 CPUs @ 2.1GHz)



**Steve Jobs Urn**

---

```
# app/config/config_dev.yml
assetic:
    use_controller: true
```

# Dev (less than 2 CPUs and 2GHz)



```
# app/config/config_dev.yml
```

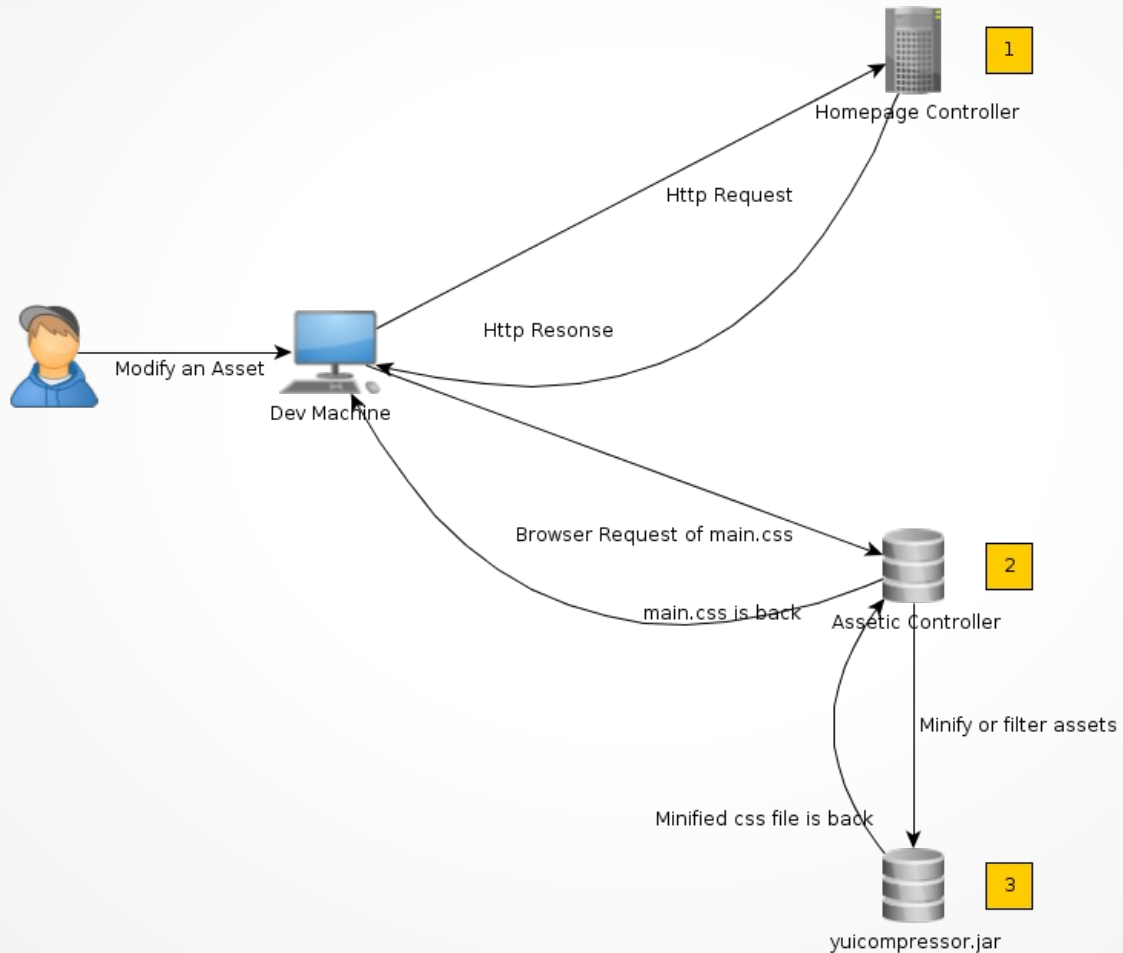
```
assetic:
```

```
    use_controller: false
```

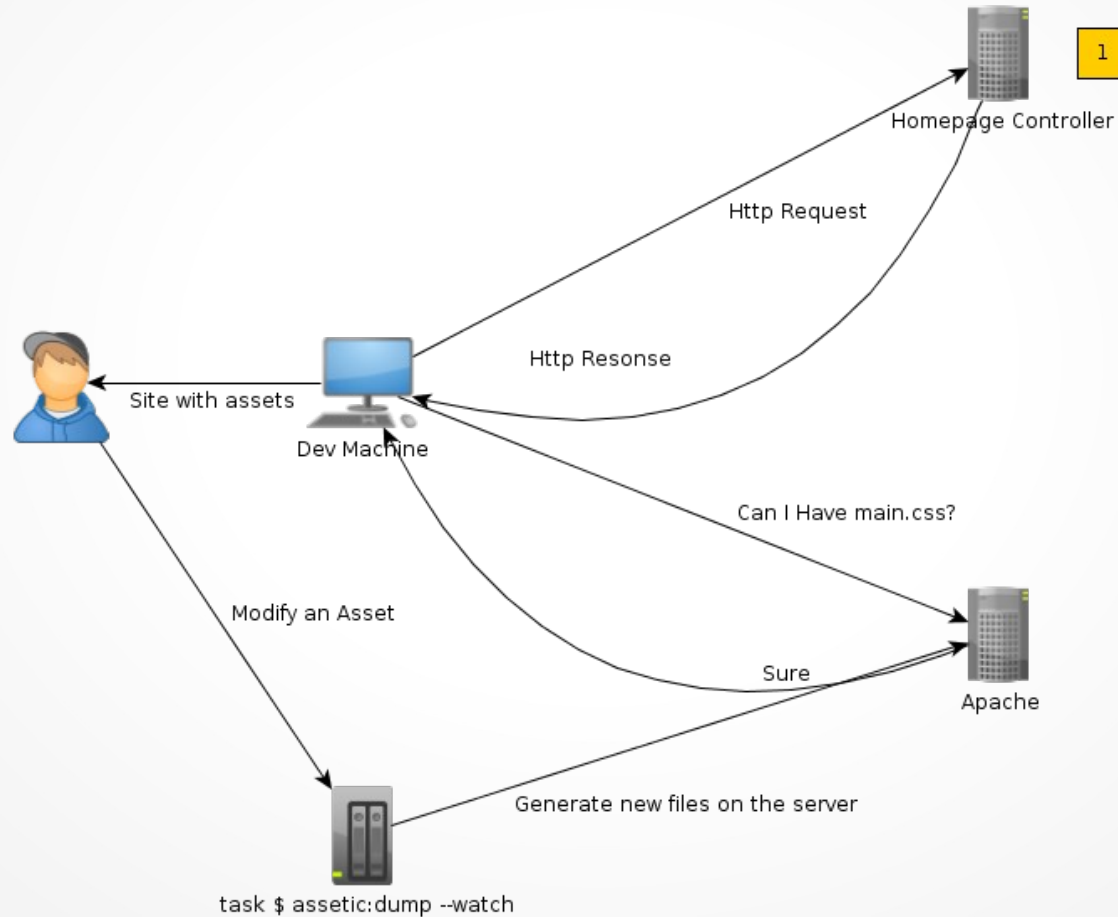
```
$ php app/console assetic:dump
```

```
$ php app/console assetic:dump  
--watch
```

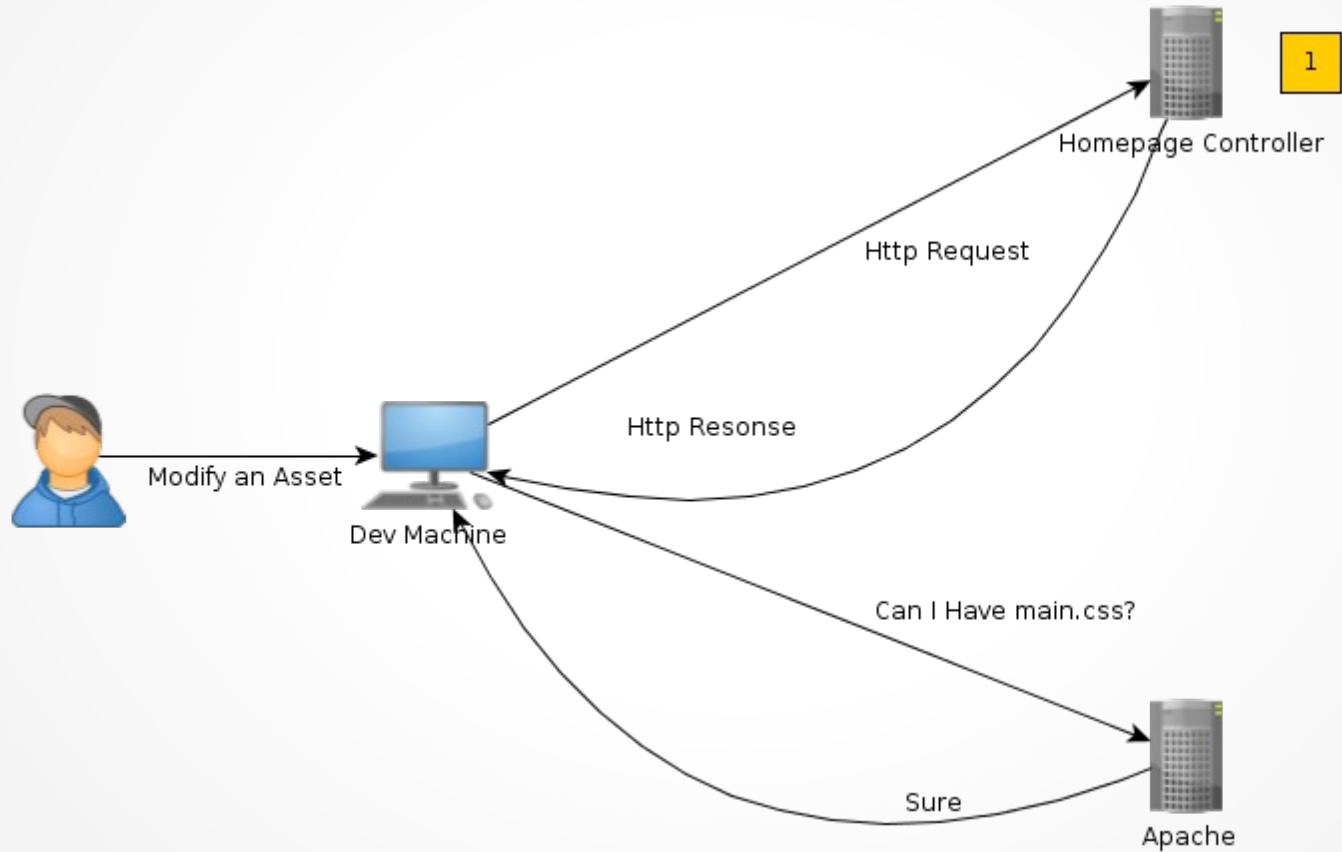
# The Dev Flow



# The Wooden PC Dev Flow



# The Prod Flow



# The Prod Flow

- All files are compiled & minified during the build
- They are copied to their destination folder
- Apache serves them as normal files.
- You can add support for CDN (Amazon S3, CloudFlare, etc.)

# CDN Support?

framework:

templating:

engines: ['twig']

assets\_base\_urls:

http: [<http://yourcdn.com>]

Yes Please.

# Q&A

- ...

**THANK YOU!**

