APACHE SLING & FRIENDS TECH MEETUP

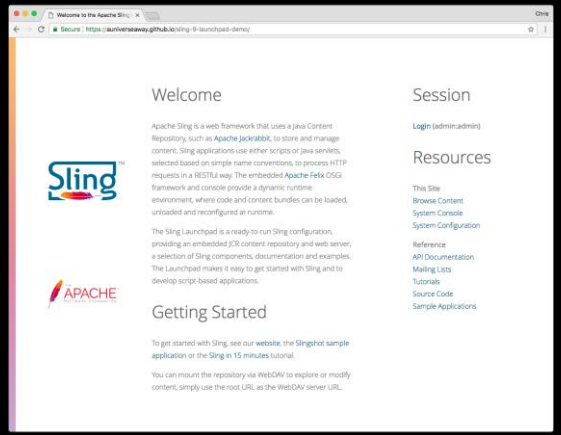
BERLIN, 25-27 SEPTEMBER 2017

Slick — Not your typical Sling Blog Engine

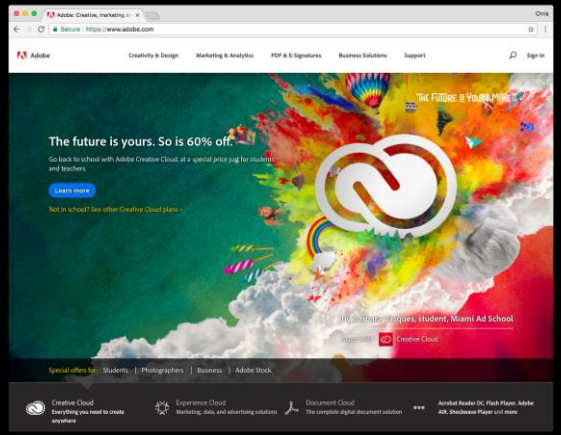
Chris Millar, Adobe Systems.

adaptTo()

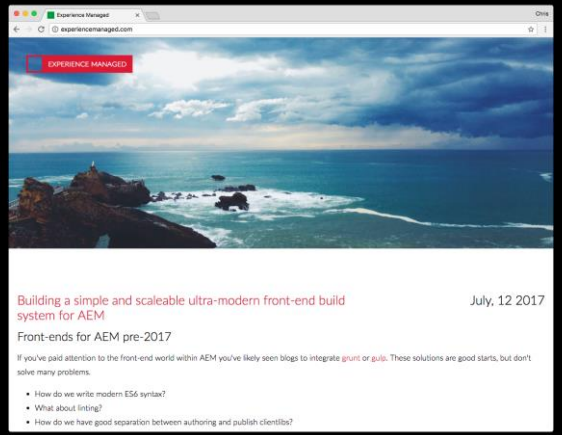
Who am I?



Sling 9 Launchpad



adobe.com



experiencemanaged.com

What is Slick?

A blog engine obsessively crafted around Apache Sling and related technologies.

(it's not a CMS)

Why was Slick built?

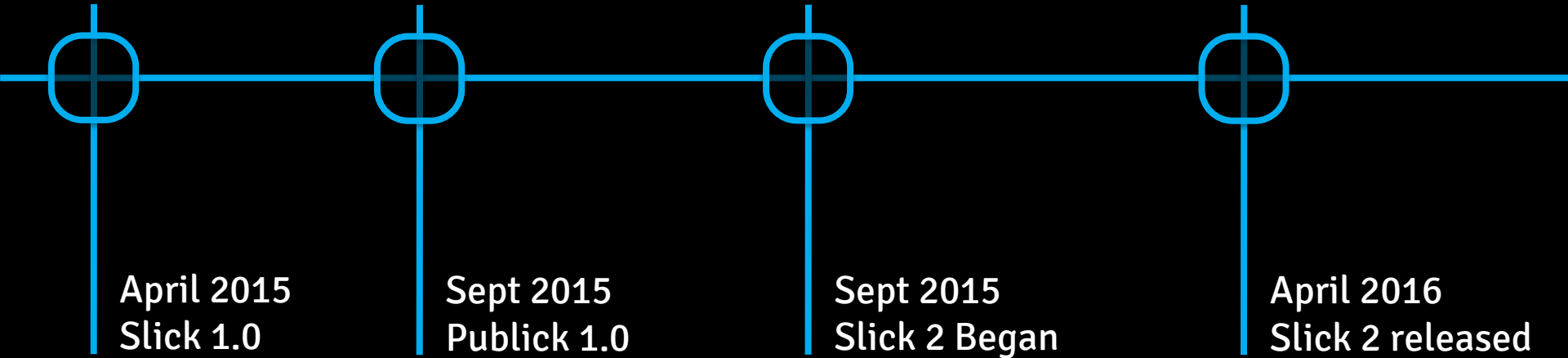
1. I had **a lot** to learn.

2. Writing a blog about a Java application using a PHP blog engine seemed wrong.

3. Sightly was contributed to Apache Sling.

adaptTo()

History



adaptTo()

What can it do? – Authors & Visitors

Easy to get creating content fast.

Creating and editing posts, pages, media, and tags.

Post and Page Scheduling

WYSIWYG Editor

RSS 2.0 Feed

Accent color support

Commenting

Authentication and user management

Search

Analytics, SEO, and social integrations

Pagination

Basic Localization

What can it do? – Developers

Proper OSGi services and configurations

Sling Models used extensively

Dispatcher support**

Entire stack written to be easy to maintain (Java, JS, HTL, SCSS)

Front-end build process is part of `autoInstallBundle`.

Uses a Maven multi-module project similar to AEM Archetype

Travis CI Integration

Basic localization string support

Eclipse Sling Plugin support

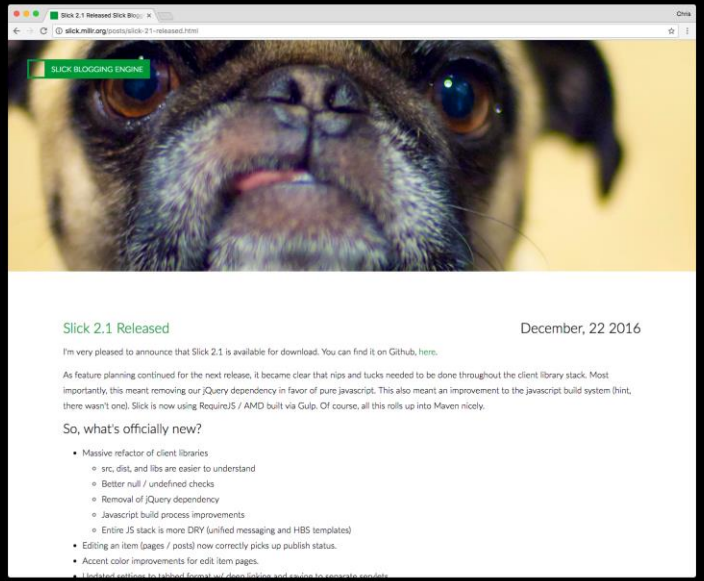
Container support

Decent Docs

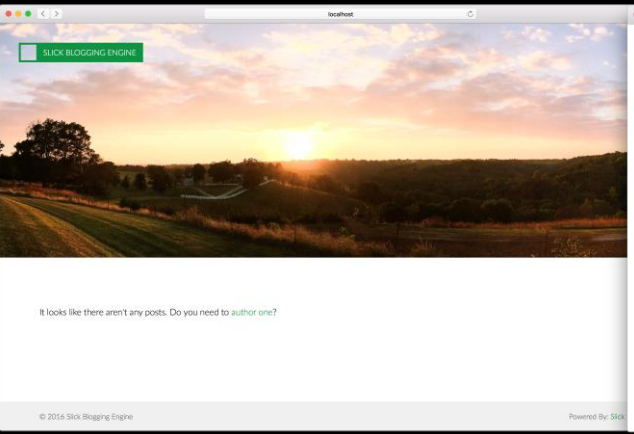
Crazy Fast On-Boarding

```
$ java -jar launchpad.jar
```

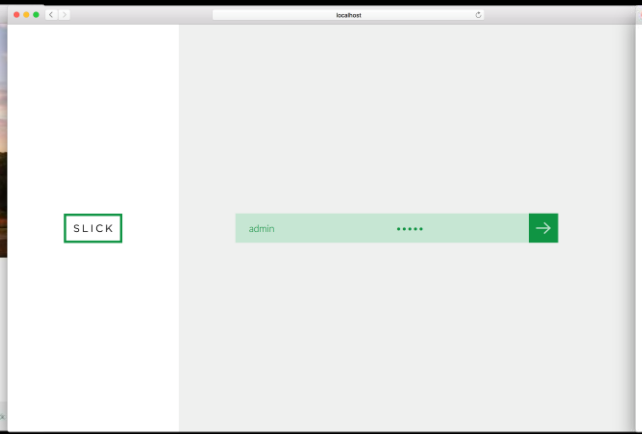
3 minutes



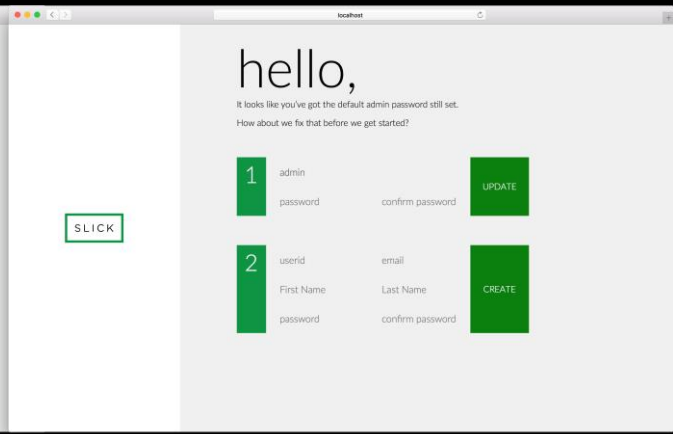
On-Boarding



1

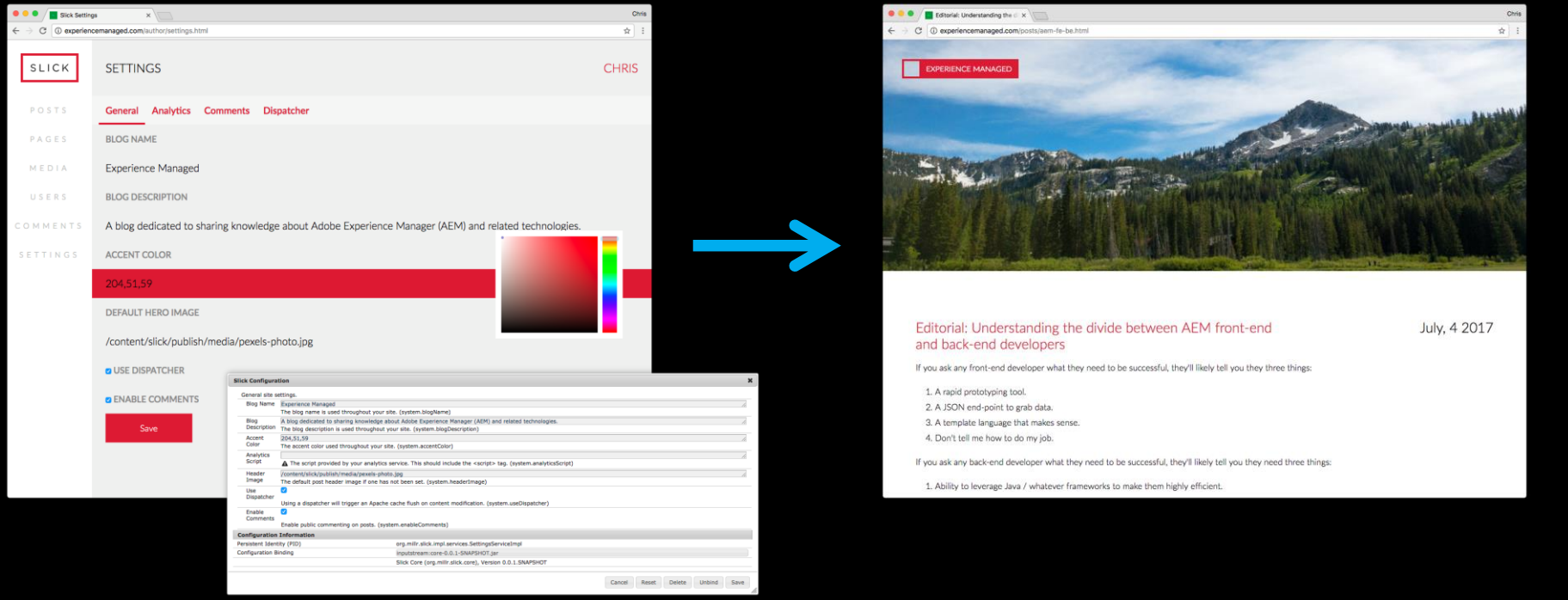


2

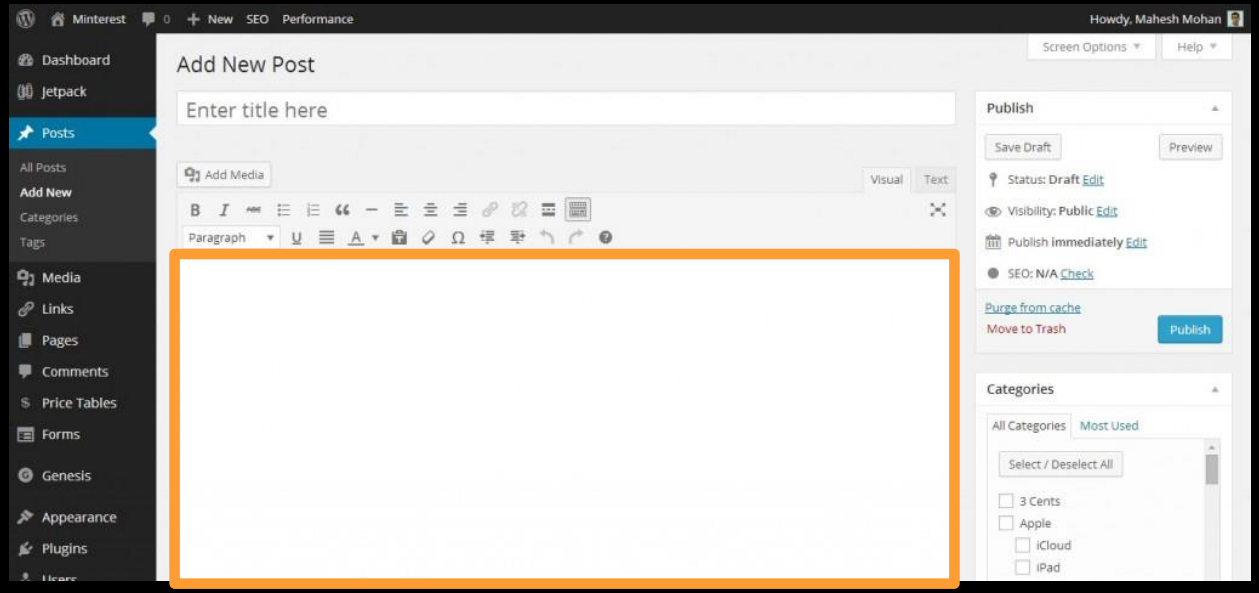


3

Settings & Themes

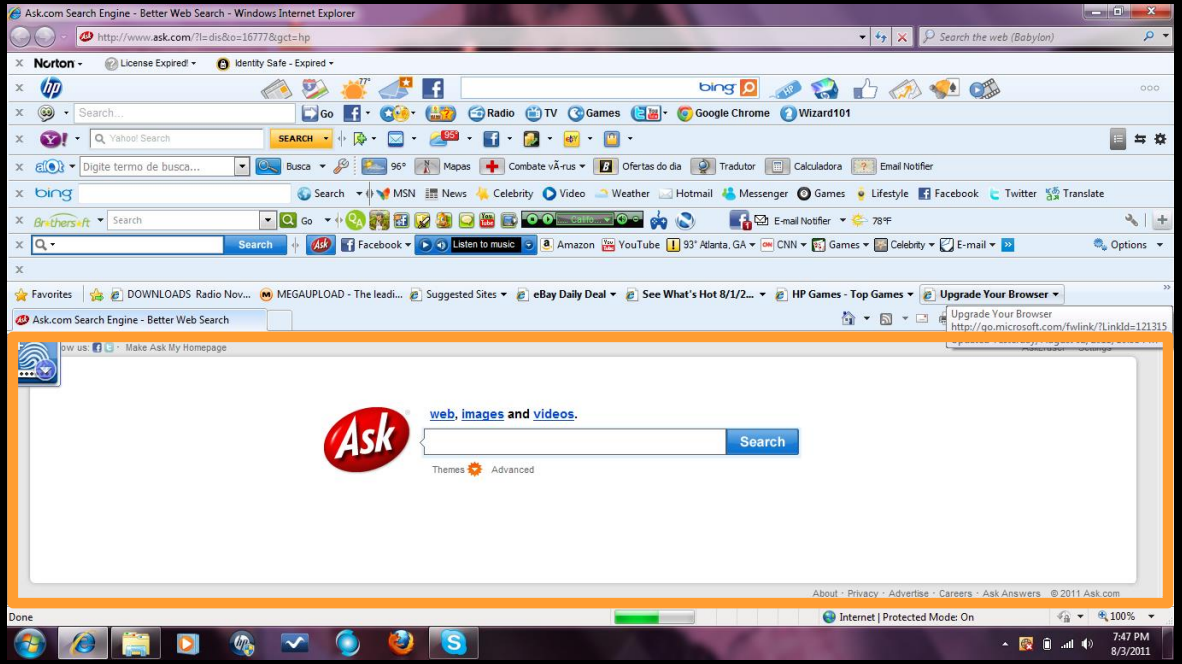


WYSIWYG Editor



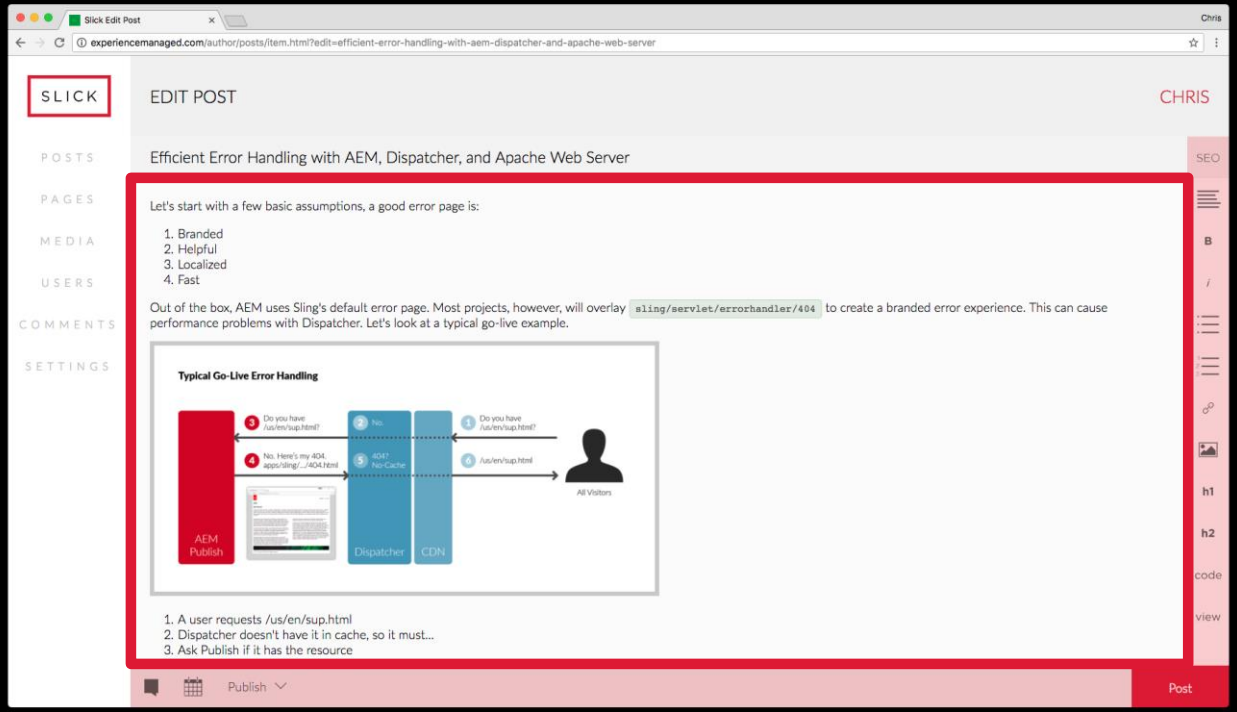
A rant

WYSIWYG Editor



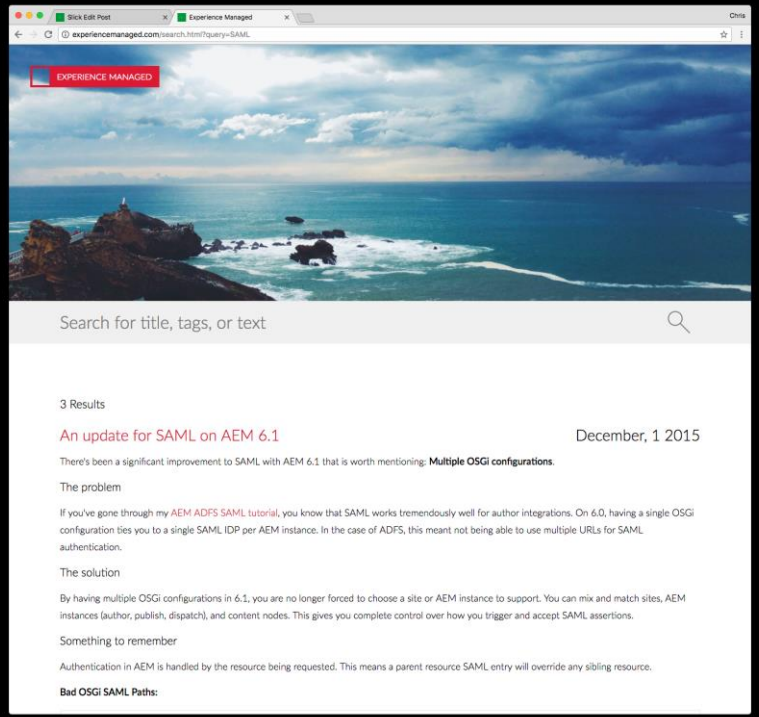
Remember this?

WYSIWYG Editor



Respect
The Content

Search



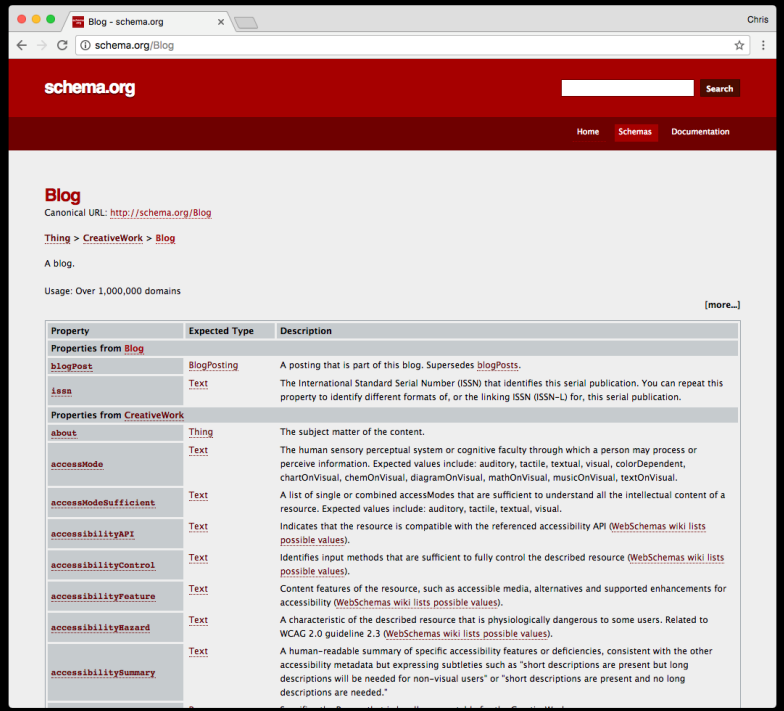
Search

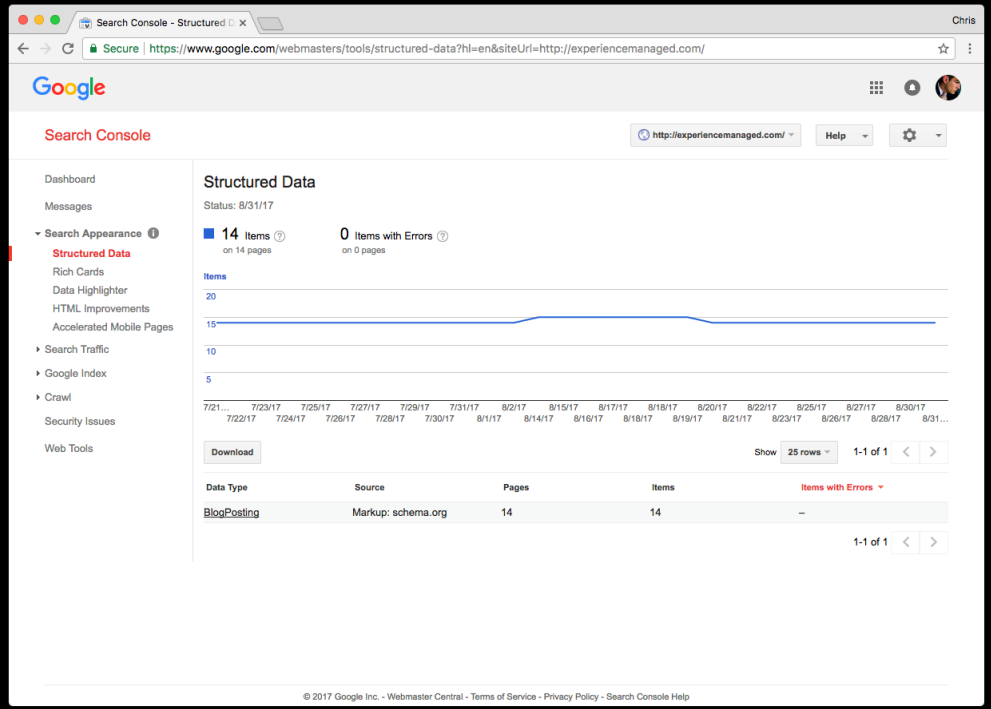
Search

```
42      private static final String INITIAL_QUERY = "SELECT "
43                                                + "* FROM [slick:page] "
44                                                + "AS s "
45                                                + "WHERE contains(s.*, '%s') "
46                                                + "AND ISCHILDNODE(s,'" + SlickConstants.POSTS_PATH + "')";
47
```

(nothing too impressive)

SEO, Social, Analytics



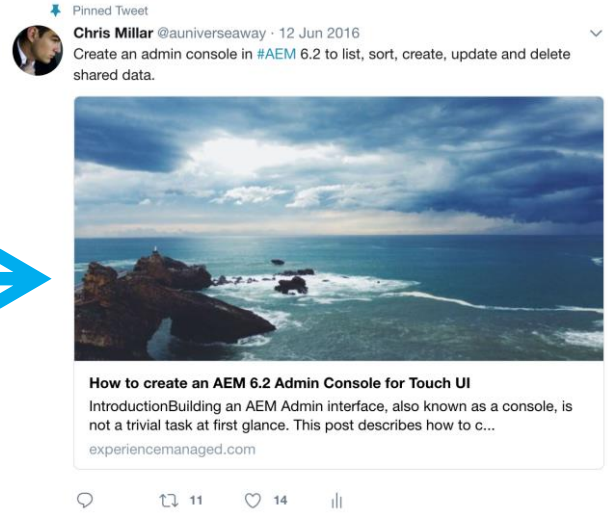


SEO, Social, Analytics

aem front-end back-end tooling webpack npm ES6

0

Chris Millar



Lightweight & Effortless Sharing

SEO, Social, Analytics

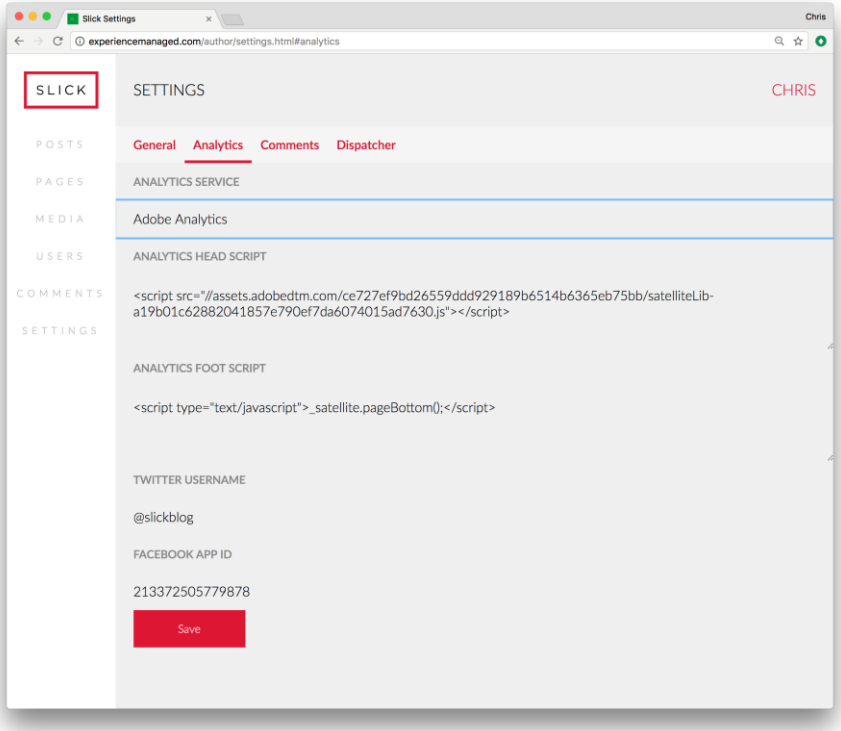
How lightweight?

55 lines (49 sloc) | 1.73 KB — Raw | Blame | History

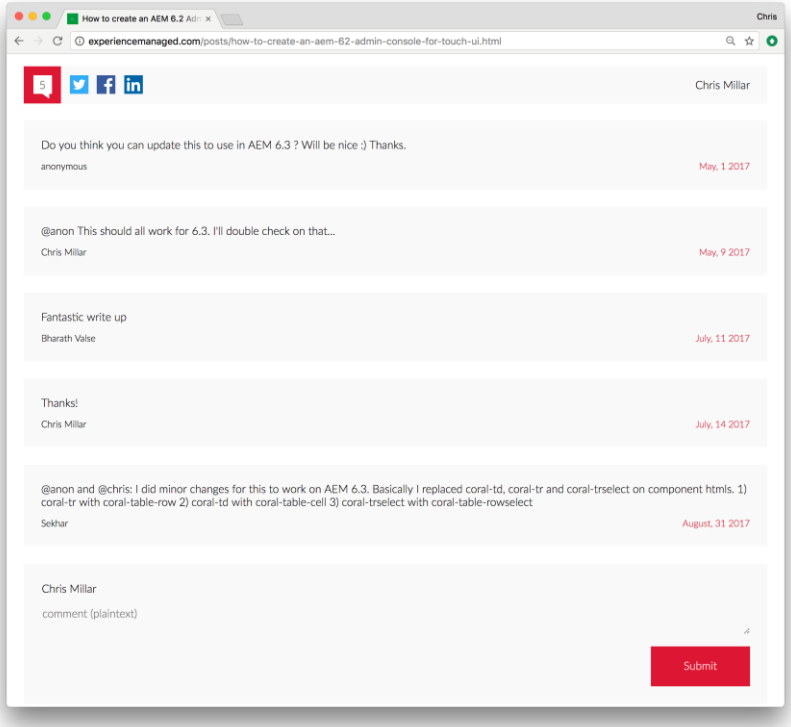
```
/*global window, define */
/**
 * Analytics, Social, and SEO Functions.
 *
 * <p>This module will handle new windows for social sharing.</p>
 * <p>If analytics are enabled, send the appropriate events to those services</p>
 */
define(function () {
    'use strict';

    var analyticsType = document.body.dataset.analyticsType;

    /**
     * Send social information to analytics services.
     * @param  {String} shareUri      The URL we're sharing.
     * @param  {String} socialNetwork The social network.
     * @return {void}
     */
    var trackLink = function (shareUri, socialNetwork) {
        if (analyticsType === 'adobeAnalytics') {
            var s = window.s_c_il[0];
            if (s) {
                s.tl(shareUri, 'o', socialNetwork, {linkTrackVars: 'prop1', prop1: s.pageName});
            }
        } else if (analyticsType === 'googleAnalytics') {
            if (window.ga) {
                window.ga('send', 'social', socialNetwork, 'Share', shareUri);
            }
        }
    };

    var forEach = function (array, callback, scope) {
        var i;
        for (i = 0; i < array.length; i += 1) {
            callback.call(scope, array[i]);
        }
    };

    var handleClick = function (element) {
        element.addEventListener('click', function (event) {
            event.preventDefault();
            var socialNetwork = this.dataset.socialNetwork;
            var shareUri = this.dataset.shareUri;
            trackLink(shareUri, socialNetwork);
            window.open(this.getAttribute('href'), socialNetwork, 'height=500,width=575');
        }, false);
    };

    // Share Windows
    var socialLinks = document.querySelectorAll('.social');
    if (socialLinks) {
        forEach(socialLinks, handleClick);
    }

});
```

SEO, Social, Analytics

Google Analytics
Adobe Analytics
DTM, GTM
Other (Piwick, Open Analytics)



Comments



So what?

Comments


```
{
  "responseCode": 200,
  "responseType": "success",
  "responseMessage": "success",
  "content": {
    "count": 5,
    "comments": [
      {
        "name": "2c45af72-9a12-44d0-92e9-db8c852832fa",
        "author": "anonymous",
        "comment": "Do you think you can update this to use in AEM 6.3 ? Will be nice :) Thanks.",
        "created": "May, 1 2017"
      },
      {
        "name": "0a66c216-6009-437c-8fe2-f3ecb6d984ce",
        "author": "Chris Millar",
        "comment": "@anon This should all work for 6.3. I'll double check on that...",
        "created": "May, 9 2017"
      },
      {
        "name": "06710120-e49b-4cbb-8756-210383cf3149",
        "author": "Bharath Valse",
        "comment": "Fantastic write up",
        "created": "July, 11 2017"
      },
      {
        "name": "14b5381f-1257-457d-bd41-a4ed16a7b5d9",
        "author": "Chris Millar",
        "comment": "Thanks!",
        "created": "July, 14 2017"
      },
      {
        "name": "259131b2-32a4-44c5-8caf-718f6e523299",
        "author": "Sekhar",
        "comment": "@anon and @chris:\r\nI did minor changes for this to work on AEM 6.3. Basically I replaced
        coral-td, coral-tr and coral-trselect on component htmls. \r\n1) coral-tr with coral-table-row\r\n2)
        coral-td with coral-table-cell\r\n3) coral-trselect with coral-table-rowselect",
        "created": "August, 31 2017"
      }
    ]
  }
}
```


Comments are JSON endpoints retrieved via AJAX to optimize for cache-ability.

adaptTo()

Architecture - Comments

/posts.html

/posts/hello-world.html

/posts/hello-world.**list.comments**.json

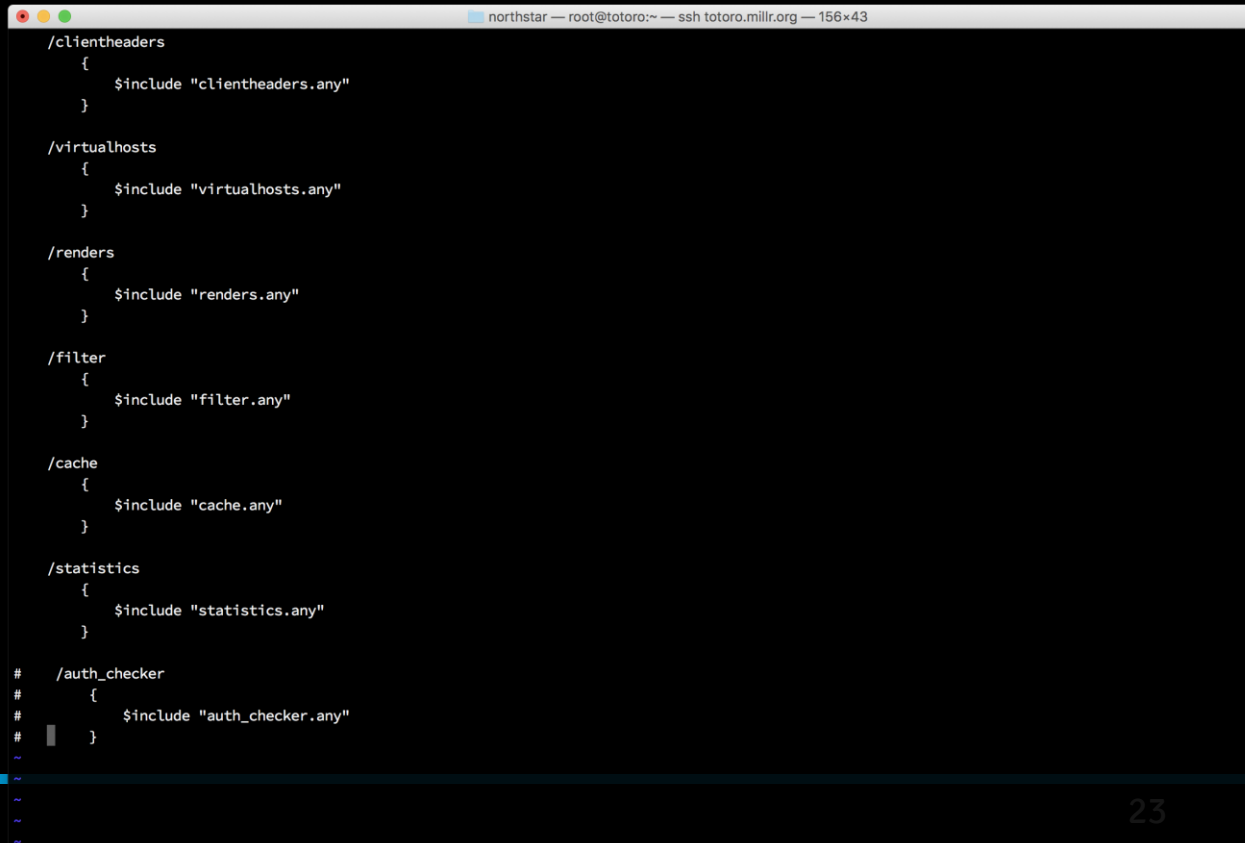
Comment Count

Comment List

Dispatcher

Dispatcher is supported

(Legal implications, read fine print)



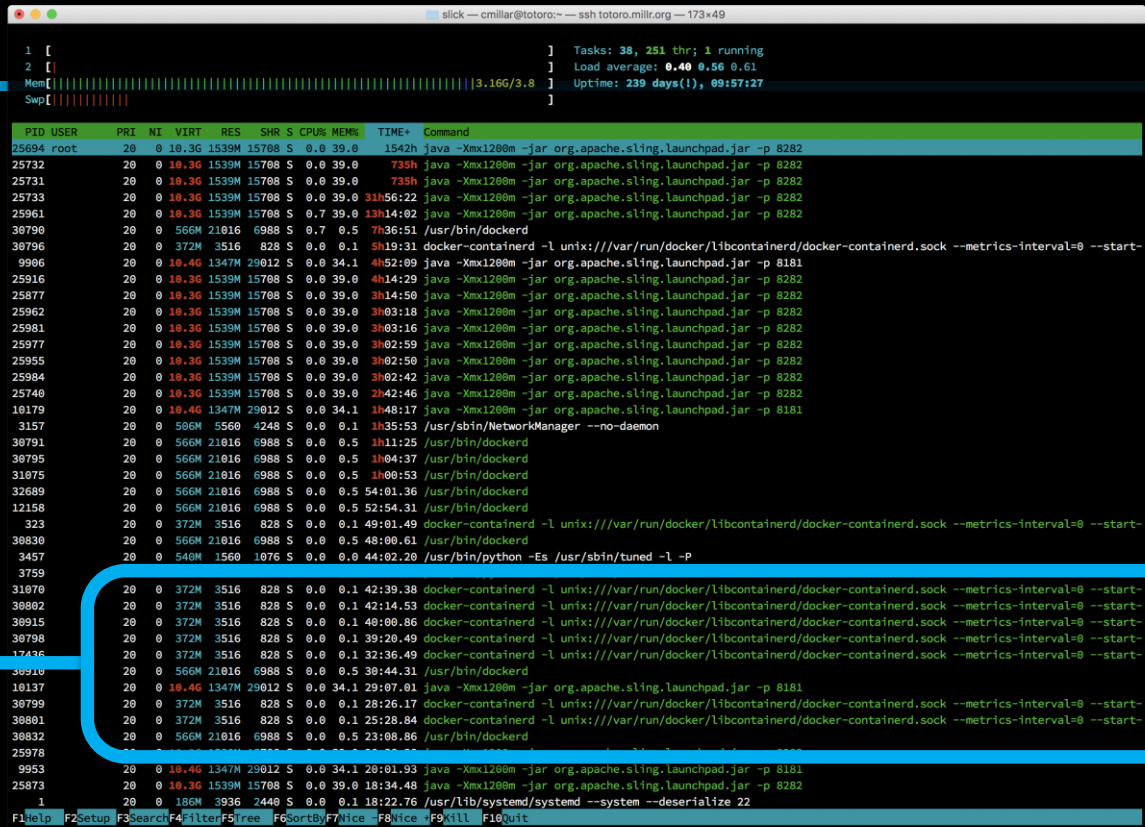
Architecture - Dispatcher Interface

```
16 references
20  public interface DispatcherService {
21
        2 references
22      Boolean canFlush();
23
        6 references
24      JSONObject flush(String domain, String flushType);
25
        1 reference
26      JSONObject flushAll(String domain);
27
        1 reference
28      JSONObject flushUi(String domain);
29
        1 reference
30      JSONObject flushContent(String domain);
31
        1 reference
32      JSONObject flushContent(String domain, String path);
33
34  }
```

Simple &
Powerful
Flush Service

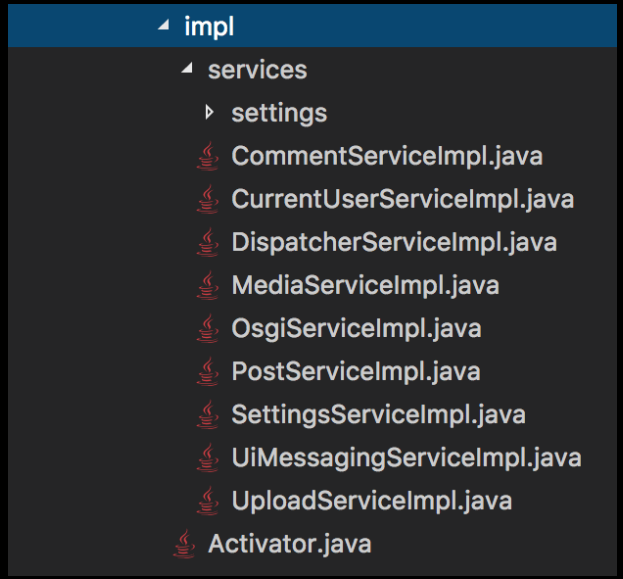
Containers

Container Support



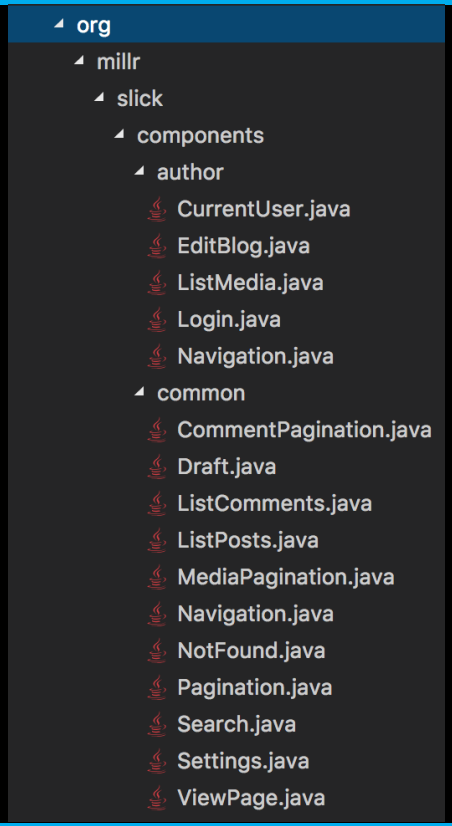
Architecture

OSGi Service
Implementations



Architecture

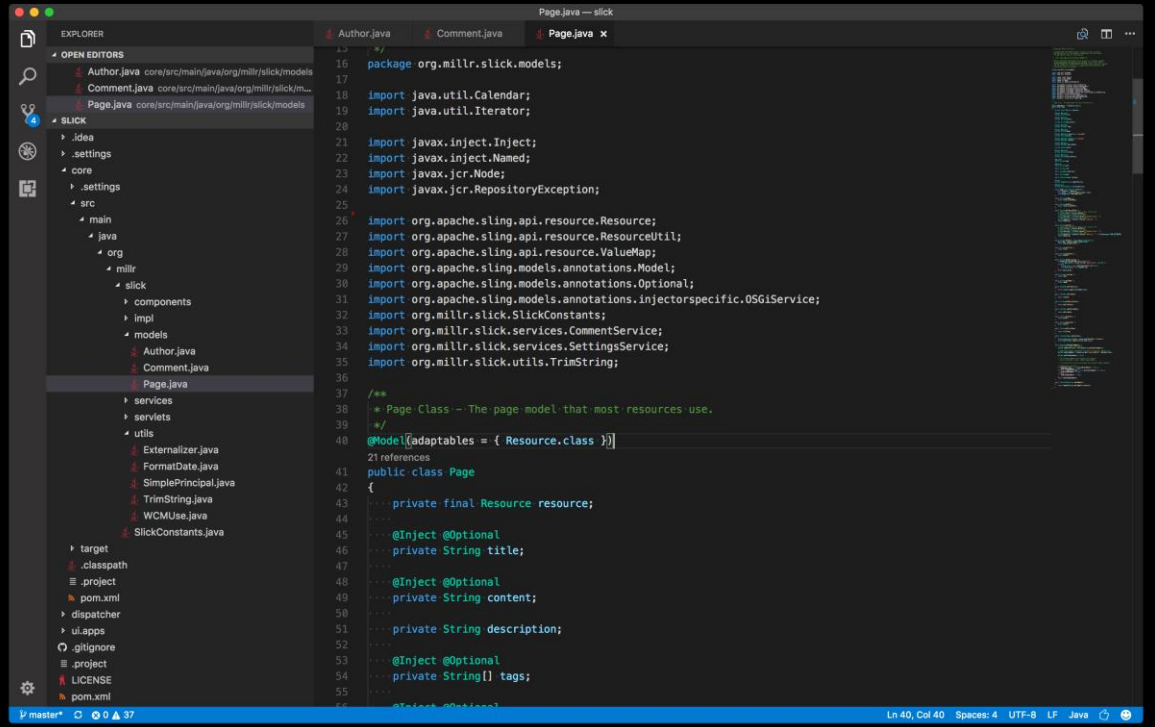
Component
Sling Models
Exposed to pages



Architecture

Base Sling Models

Page
Author
Comment



Architecture - HTL

```
${view.page.title}

${view.page.body}

${view.author.firstName}

${view.external.url}
```

Straight-forward
Naming conventions

Extensive use of
data-sly-template

Architecture - HTL

```
${ view . page . author . firstName }
```

View.java Page.java Author.java getFirstName()

Architecture - Page.java

```
194     public Iterator<Page> getChildren()
195     {
196         Iterator<Resource> children = resource.getChildren().iterator();
197         return ResourceUtil.adaptTo(children, Page.class);
198     }
199
```

Demo

Contact & Links

Slick Source	http://github.com/auniverseaway/slick-2
Twitter	http://twitter.com/auniverseaway (a universe away)
Slick Blog	http://slick.millr.org

Q & A