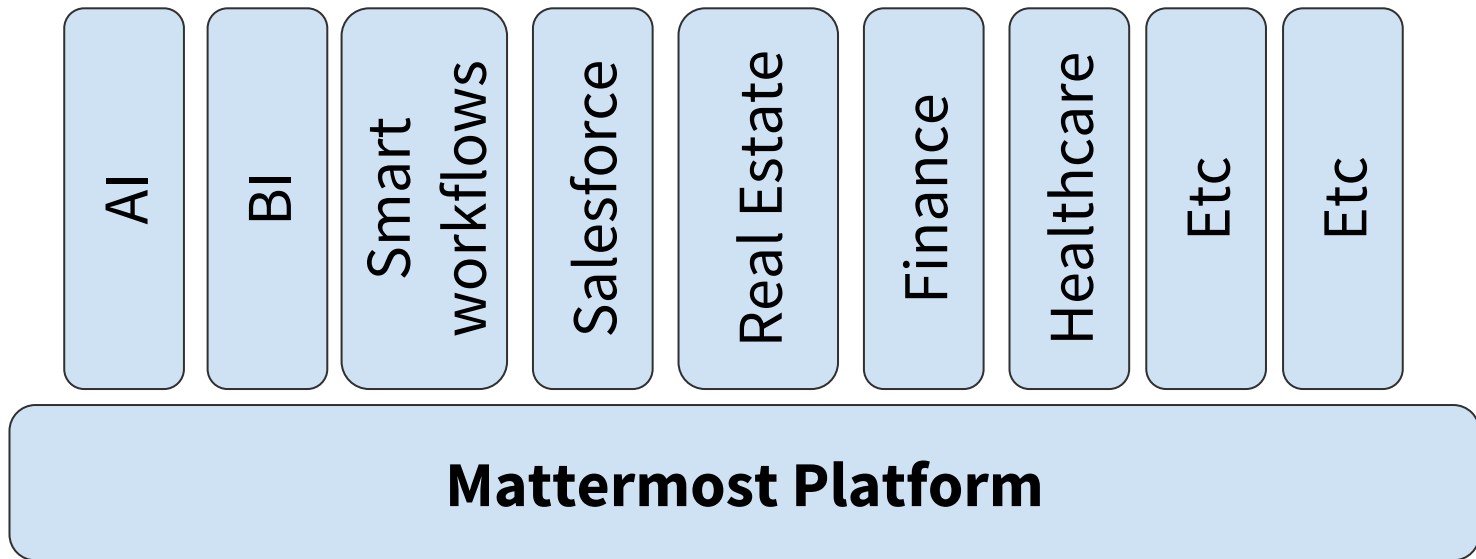


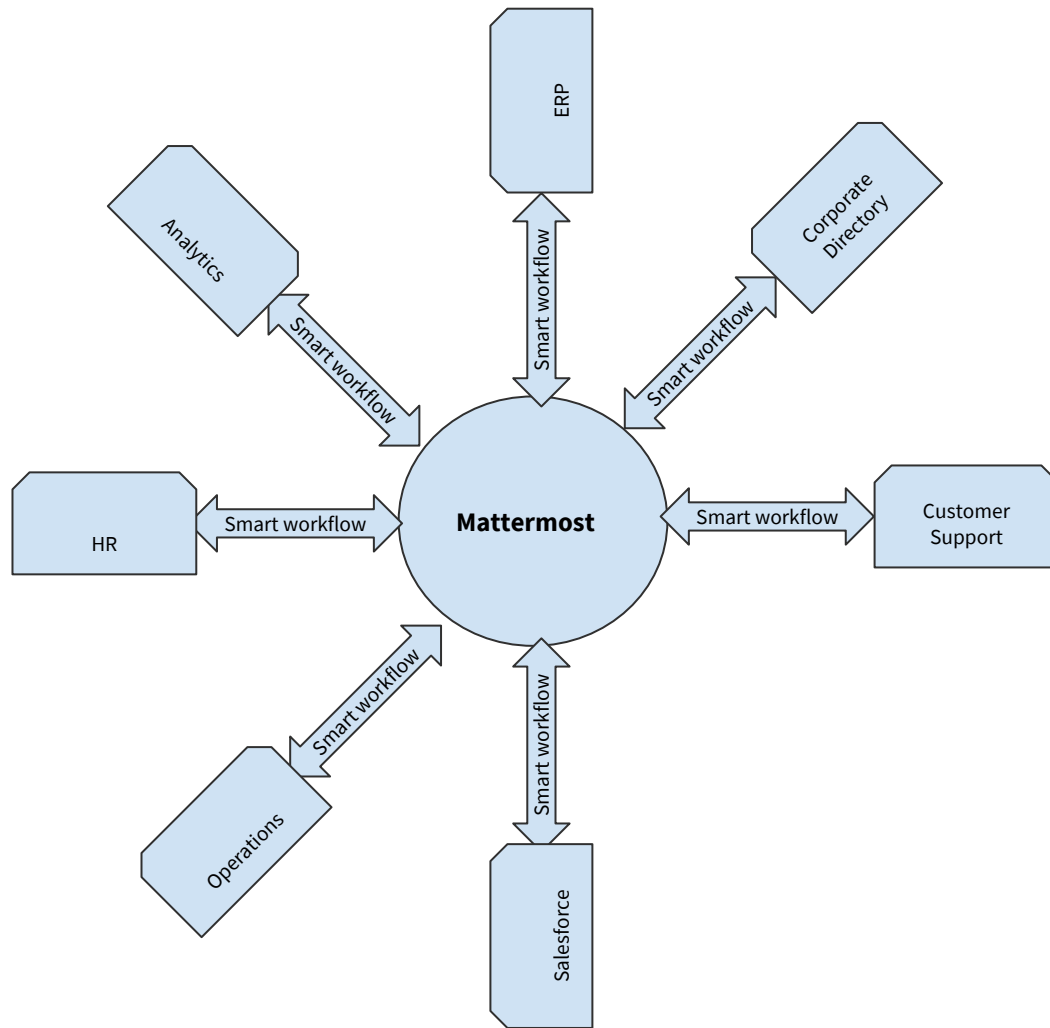
Mattermost's Approach to Layered Extensibility in Open Source

Corey Hulen / Co-founder and CTO, Mattermost, Inc.
Feb 2019

Mattermost as an OS

- Mattermost as a platform from the very beginning





Embedding Applications
inside of Mattermost

Mattermost has Layers!

Slash Commands and Webhooks

REST API/Bots

Plugins

Developer
Toolkit

Open Source

Open Data

Mattermost is Open Source

- Typical open source goodness
- Fork, change what you want (heck change everything!)
- Hard, takes a lot of resources
- People love the idea but in reality very rarely happens
- 100% of what you see in the UI is accessible from our RESTful JSON web service API

Open Source Example

Matterhorn is a terminal client for the Mattermost chat system.

<https://github.com/matterhorn-chat/matterhorn>

```
Channels
#comics
#matterhorn
#mh-git
#minecraft
#music
#mutt
#off-topic
#scratch
#tmux
#town-square
#vim
Users
+aaron
-acoltzer
-amaloz
+andrey
-conathan
-diatckki
+dmwit
-dmz
+dwest
+gdritter
-gerit
-huffman
+ijones
-kiniry
-max.orhai
+mpodhradsky
+tommd
-trevor
+tristan
+westbrook
anna

#music - Music!
everyone. Watch till the end)
• https://www.youtube.com/watch?v=LWGJA9i18Co (filmed in the plane that gives
you a bit of 0 gravity)
The making of videos are also awesome and informative.
[10:26] dagit: (also, I don't think Lee appreciated (understood?) my joke the other day that
the hacms build system should be featured in an ok go video)
[10:56] trevor: i've been really liking moka only recently:
https://urbnet.bandcamp.com/album/carrots-and-eggs
2016-12-16
[11:12] rdockins: Chiptunes! https://slimegirls.bandcamp.com/album/vacation-wasteland-ep
[13:22] dlamkins: If you like chiptune, here's some from my friend Matt Hunter:
https://mechlo.bandcamp.com/
[13:32] dlamkins: Also, my trio: https://lcwmusic.bandcamp.com/
[13:34] dlamkins: And my friend Ben Spees' prog/jazz band:
https://themercurytree.bandcamp.com/
[13:37] dagit: Not strictly chiptunes (but heavily influenced by) virt is one of my favorite
musicians on bandcamp: https://virt.bandcamp.com/
[13:37] dagit: This take on the crypto of the necrodancer music is really good for when
you're getting in the groove on some programming.
[13:37] dagit: er, his take
[13:39] rdockins: crypto of the necrodancer sounds like it would be pretty rad stream cypher
[13:53] rdockins: thanks @dlamkins, good stuff!
[14:09] dlamkins: Portland metal: https://redfang.bandcamp.com/
[14:10] bjones: redfang is great
[14:13] dlamkins: Yeah! I've seen them several times at small venues and once at a packed
show at Wonder Ballroom. Their music videos are great, too. My favorite
one: https://www.youtube.com/watch?v=WQPfQvLIseA
[14:25] bjones: haha 80 gallons of milk
[14:28] bjones: that was amazing
2016-12-19
[11:19] rdockins: @dlamkins, that video is great! I especially liked the piano.
[11:20] rdockins: I was introduced to Sleigh Bells this weekend; they're a fun noise-pop duo.
[11:20] rdockins: https://sleighbells.bandcamp.com
```

Mattermost is Open Data

- You own 100% of your data
- Easy extraction and manipulation with SQL
- Easily chart, report, analyze
- No proprietary data models; nothing hidden

Developer Toolkit

- Incoming/Outgoing Webhooks
- Slash Commands
- Bots
- REST API
- Client-side Plugins
- Server-side Plugins

<https://developers.mattermost.com>

Webhooks

- **Incoming Webhooks**

- Post messages to public, private and direct message channels
- Simple HTTP POST request to a URL
- Designed to easily allow you to post messages
- Slack compatible

- **Outgoing Webhooks**

- Post automated responses to posts made by your users
- Uses trigger words
- Simple HTTP POST request to a web service
- Slack compatible

Webhooks Example

 alert 11:48 AM 

...   2

ALARM: "corp-mattermost-CPU-Utilization" in US East (N. Virginia)

[#aws-notification-alert](#) on [Sat, 3 Nov 2018 18:45:57 +0000](#)

You are receiving this email because your Amazon CloudWatch Alarm "corp-mattermost-CPU-Utilization" in the US East (N. Virginia) region has entered the ALARM state, because "Threshold Crossed: 1 datapoint [82.4641844956938 (03/11/18 18:40:00)] was greater than or equal to the threshold (50.0)." at "Saturday 03 November, 2018 18:45:57 UTC".

View this alarm in the AWS Management Console:
<https://console.aws.amazon.com/cloudwatch/home?region=us-east-1#s=Alarms&alarm=corp-mattermost-CPU-Utilization>

Alarm Details:



github 07:10

PLT-6905 - Updating channel header design

[#new-comment](#) #6789 in [mattermost/platform](#) by Hanzei on [2017-07-06T11:09:35Z](#)

On the Spinmint server clicking on "Recent Mentions" does nothing. Is this intended?

Slash Commands

- Easy way integrate external applications
- Similar to outgoing webhooks, but can be used in any channel
- Simple HTTP POST request to a URL, process response back into channel
- Both built-in and custom slash commands
- Slack compatible

Matterbuild example



cpanato 21:18

ok

for 5.5: [pr-mattermost-server-9799](#)

for master: [pr-mattermost-server-9798](#)

```
/mb cut 5.5.0-rc1
```



Matterbuild BOT 11:12

Matterbuild

Cut Release

Release 5.5.0-rc1 is on the way.



Mattermost Release BOT 11:15

Release 5.5.0-rc1 has started building.

Bots

- Official and community-built drivers for our web service API
 - Javascript
 - Golang
 - PHP
 - Python
 - <https://api.mattermost.com>
- Golang bot sample code
 - Uses the go driver to interact with the Mattermost server
 - Login to the server, create a channel, modify a user, etc
 - <https://github.com/mattermost/mattermost-bot-sample-golang>

Sample Bot Example

```
98 func LoginAsTheBotUser() {
99     if user, resp := client.Login(USER_EMAIL, USER_PASSWORD); resp.Error != nil {
100         println("There was a problem logging into the Mattermost server. Are you sure ran the setup steps from the README.md?")
101         PrintError(resp.Error)
102         os.Exit(1)
103     } else {
104         botUser = user
105     }
106 }

161 func SendMsgToDebuggingChannel(msg string, replyToId string) {
162     post := &model.Post{}
163     post.ChannelId = debuggingChannel.Id
164     post.Message = msg
165
166     post.RootId = replyToId
167
168     if _, resp := client.CreatePost(post); resp.Error != nil {
169         println("We failed to send a message to the logging channel")
170         PrintError(resp.Error)
171     }
172 }
```

Hubot Example



Plugins Overview

- **Customize User Interfaces**
 - Write a web app plugin to add to the channel header, sidebars, main menu and more. Register your plugin against a post type to render custom posts or wire up a root component to build an entirely new experience. All this is possible without having to fork the source code and rebase on every Mattermost release.
- **Launch Tightly Integrated Services**
 - Launch and manage server plugins as services from your Mattermost server over RPC. Handle events via real-time hooks and invoke Mattermost server methods directly using a dedicated plugin API.
- **Extend the Mattermost REST API**
 - Extend the Mattermost REST API with custom endpoints for use by web app plugins or third-party services. Custom endpoints have access to all the features of the standard Mattermost REST API, including personal access tokens and OAuth 2.0.

Customize User Interfaces (Client Side)

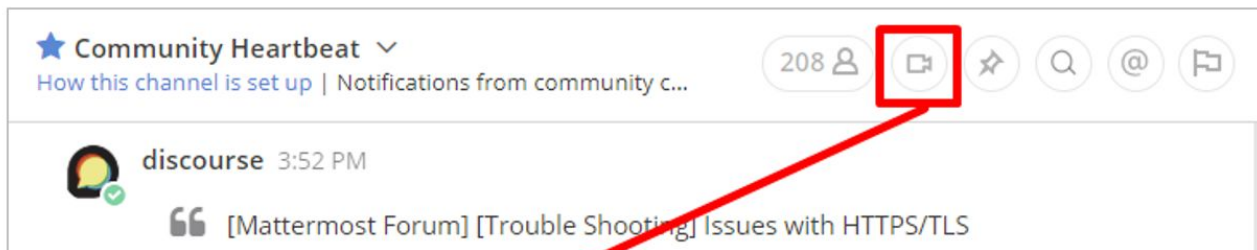
- `registerRootComponent(component)`
- `registerPopoverUserAttributesComponent(component)`
- `registerPopoverUserActionsComponent(component)`
- `registerLeftSidebarHeaderComponent(component)`
- `registerBottomTeamSidebarComponent(component)`
- `registerPostMessageAttachmentComponent(component)`
- `registerChannelHeaderButtonAction(icon, action, dropdownText, tooltipText)`
- `registerPostTypeComponent(type, component)`
- `registerMainMenuAction(text, action, mobileIcon)`
- `registerPostDropdownMenuAction(text, action, filter)`
- `registerPostDropdownMenuComponent(component)`
- `registerFileUploadMethod(icon, action, text)`
- `registerFilesWillUploadHook(hook)`
- `unregisterComponent(componentId)`
- `registerReducer(reducer)`
- `registerWebSocketEventHandler(event, handler)`
- `unregisterWebSocketEventHandler(event)`
- `registerReconnectHandler(handler)`
- `unregisterReconnectHandler()`
- `registerMessageWillFormatHook(hook)`

Channel Header Button Example

- Add a button to the channel header
- If there is more than one button, a dropdown menu is created
- Great for things like video conferencing integration
- Our built-in Zoom integration is actually a plugin

```
14   initialize(registry, store) {  
15     registry.registerChannelHeaderButtonAction(  
16       <Icon/>,  
17       (channel) => {  
18         startMeeting(channel.id)(store.dispatch, store.getState);  
19       },  
20       'Start Zoom Meeting'  
21     );
```

Zoom Integration adding a Header Button



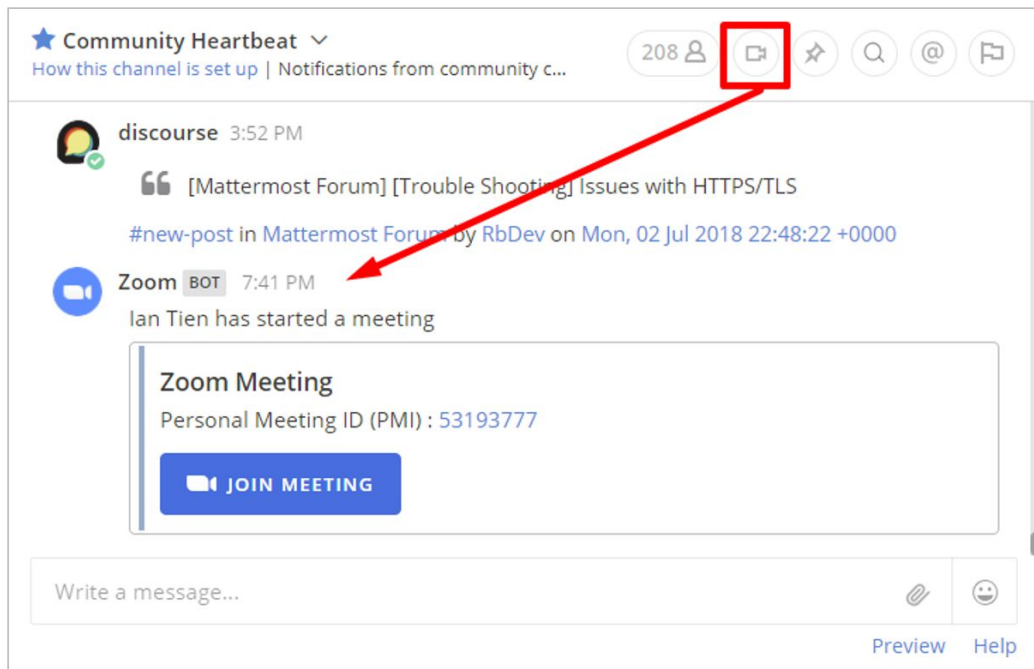
Override Post Rendering

- Register a component to render a custom body for posts
- Great for rich integrations with custom applications
- More than simple text or a screenshot
- The Zoom plugin uses this to render a custom post

```
14 initialize(registry, store) {  
15     registry.registerChannelHeaderButtonAction(  
16         <Icon/>,  
17         (channel) => {  
18             startMeeting(channel.id)(store.dispatch, store.getState);  
19         },  
20         'Start Zoom Meeting'  
21     );  
22     registry.registerPostTypeComponent('custom_zoom', PostTypeZoom);  
23 }
```

```
12 export default class PostTypeZoom extends React.PureComponent {  
13     static propTypes = {  
59     render() {  
60         const style = getStyle(this.props.theme);  
61         const post = this.props.post;  
62         const props = post.props || {};  
63  
64         let preText;  
65         let content;  
66         let subtitle;  
67         if (props.meeting_status === 'STARTED') {  
68             preText = `${this.props.creatorName} has started a meeting`;  
69             content = (
```

Zoom Integration custom rendering



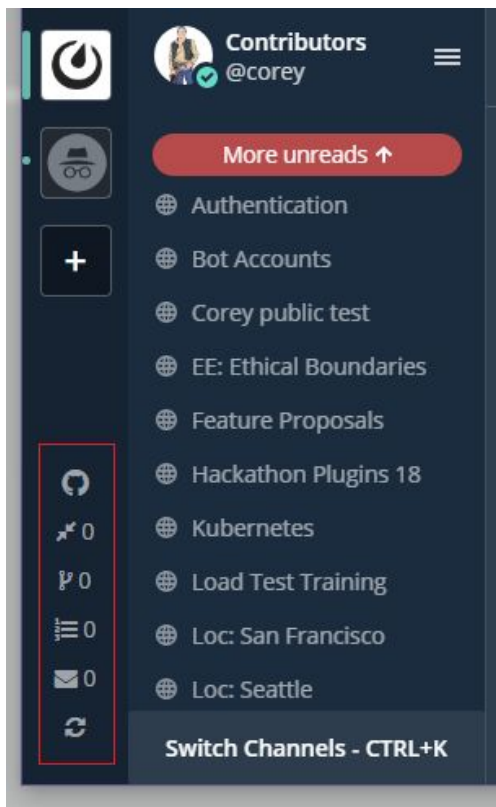
Render Component on the Team Sidebar

- Register a component fixed to the bottom of the team sidebar
- Used in our Github plugin

```
15 class PluginClass {
16   async initialize(registry, store) {
17     registry.registerReducer(Reducer);
18
19     await getConnected(true)(store.dispatch, store.getState);
20
21     registry.registerLeftSidebarHeaderComponent(SidebarHeader);
22     registry.registerBottomTeamSidebarComponent(TeamSidebar);
23     registry.registerPopoverUserAttributesComponent(UserAttribute);
24   }
```

```
6 export default class TeamSidebar extends React.PureComponent {
7   static propTypes = {
8     show: PropTypes.bool.isRequired,
9     theme: PropTypes.object.isRequired,
10  };
11
12  render() {
13    if (!this.props.show) {
14      return null;
15    }
16
17    return (
18      <SidebarButtons
19        theme={this.props.theme}
20        isTeamSidebar={true}
21      />
22    );
23  }
```

Github Integration



Rewrite the Message Client Side

- Register a hook that will be called before a message is formatted into Markdown
- Used in the Waltime plugin

```
12 export default class Plugin {
13   // eslint-disable-next-line no-unused-vars
14   initialize(registry, store) {
15     registry.registerMessageWillFormatHook((post) => {
16       const {message} = post;
17       const state = store.getState();
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70       const {text} = nlpResult;
71       newMessage = `${newMessage.replace(text, `\\`${text}\\` *(${formattedDisplayDate})*)}`;
72     }
73
74     return newMessage;
```


Walltime Plugin

Example

User in the timezone `America\Toronto` post the following message:

```
Let's meet today at 10am
```

A user in another timezone, ie. `Europe\Berlin` will see:

```
Let's meet today at 10am (Thursday, August 16, 2018 4:00 PM CEST)
```

Another example:

```
Let's meet today at noon
```

the user reading the message in another timezone will see:

```
Let's meet today at noon (Thursday, August 16, 2018 7:00 AM CEST)
```

Tightly Integrated Services (Server Side)

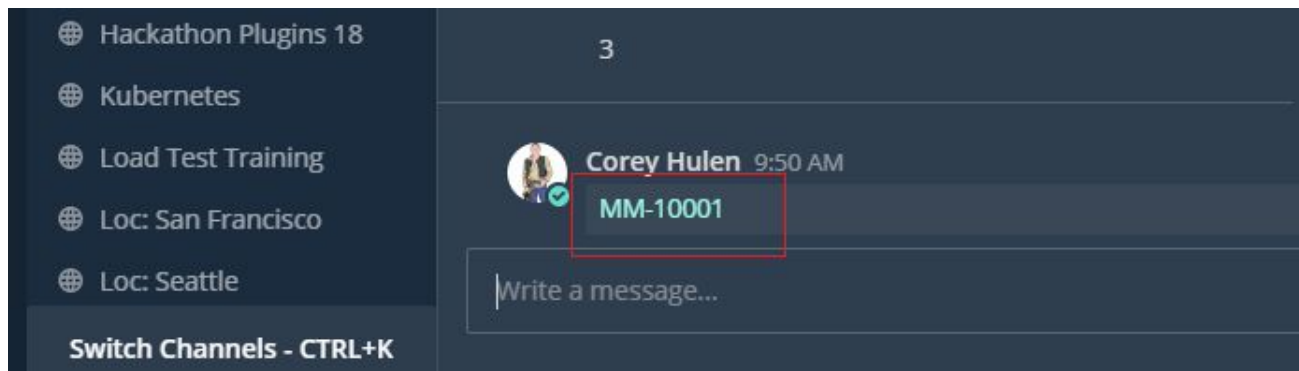
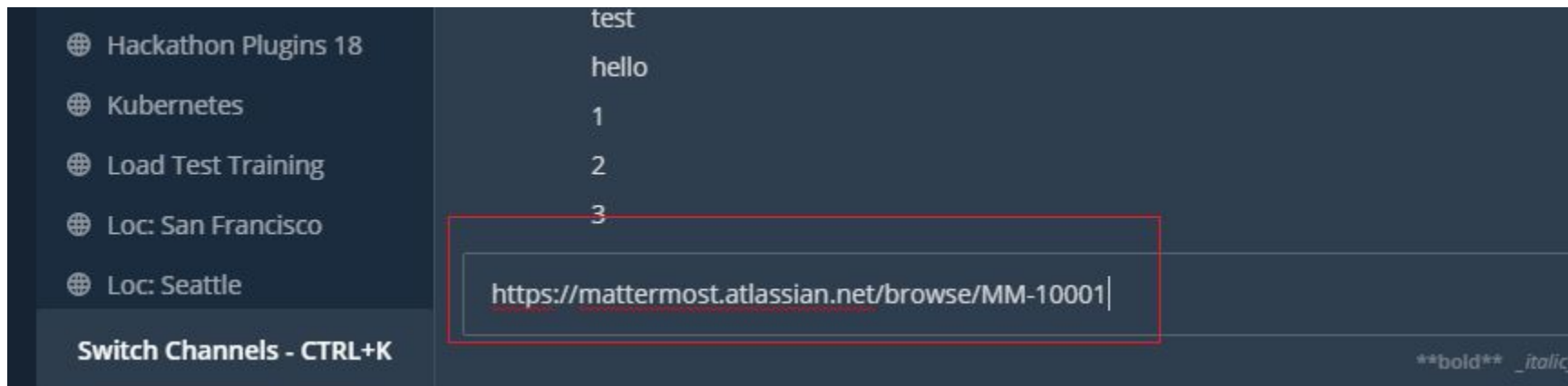
- `OnActivate() error`
- `Implemented() ([]string, error)`
- `OnDeactivate() error`
- `OnConfigurationChange() error`
- `ServeHTTP(c *plugin.Context, w http.ResponseWriter, r *http.Request)`
- `ExecuteCommand(c *plugin.Context, args *model.CommandArgs) (*model.CommandResponse, *model.AppError)`
- `UserHasBeenCreated(c *plugin.Context, user *model.User)`
- `UserWillLogIn(c *plugin.Context, user *model.User) string`
- `UserHasLoggedIn(c *plugin.Context, user *model.User)`
- `MessageWillBePosted(c *plugin.Context, post *model.Post) (*model.Post, string)`
- `MessageWillBeUpdated(c *plugin.Context, newPost, oldPost *model.Post) (*model.Post, string)`
- `MessageHasBeenPosted(c *plugin.Context, post *model.Post)`
- `MessageHasBeenUpdated(c *plugin.Context, newPost, oldPost *model.Post)`
- `ChannelHasBeenCreated(c *plugin.Context, channel *model.Channel)`
- `UserHasJoinedChannel(c *plugin.Context, channelMember *model.ChannelMember, actor *model.User)`
- `UserHasLeftChannel(c *plugin.Context, channelMember *model.ChannelMember, actor *model.User)`
- `UserHasJoinedTeam(c *plugin.Context, teamMember *model.TeamMember, actor *model.User)`
- `UserHasLeftTeam(c *plugin.Context, teamMember *model.TeamMember, actor *model.User)`
- `FileWillBeUploaded(c *plugin.Context, info *model.FileInfo, file io.Reader, output io.Writer) (*model.FileInfo, string)`

Rewrite the Message Server Side

- Lots of opportunities to hook the message
- Before it enters into the database
- You can reject messages
- You can also send back ephemeral messages
- After it enters into the database

```
46 func (p *Plugin) MessageWillBePosted(c *plugin.Context, post *model.Post) (*model.Post, string) {
47     links := p.links.Load().([]*AutoLinker)
48     postText := post.Message
49     offset := 0
50     markdown.Inspect(post.Message, func(node interface{}) bool {
51         switch node.(type) {
52             // never descend into the text content of a link/image
53             case *markdown.InlineLink:
54                 return false
55             case *markdown.InlineImage:
56                 return false
57             case *markdown.ReferenceLink:
58                 return false
59             case *markdown.ReferenceImage:
60                 return false
61         }
62     })
```

Autolink Plugin



Implement a http.Handler on the Server

- ServeHTTP allows the plugin to implement the http.Handler interface.
- Requests destined for the /plugins/{id} path will be routed to the plugin.

```
15 func (p *Plugin) ServeHTTP(c *plugin.Context, w http.ResponseWriter, r *http.Request) {
16     var action *Action
17     json.NewDecoder(r.Body).Decode(&action)
18
19     if action == nil {
20         encodeEphemeralMessage(w, "WelcomeBot Error: We could not decode the action")
21         return
22     }
23
24     data := &MessageTemplate{}
25     var err *model.AppError

```



```
76 func (p *Plugin) handleSupport(action *Action, data *MessageTemplate) {
77     p.joinChannel(action, "peer-to-peer-help")
78     p.joinChannel(action, "bugs")
79
80     template, _ := template.New("Response").Parse(
81         `### Awesome, I have added you to the following support related channels!
82 ~peer-to-peer-help - General help and setup
83 ~bugs - To help investigate or report bugs
84 `)
```

WelcomeBot Handling Custom Actions

In addition to the basics discussed above the [Contributor Code of Conduct](#) also applies.

Updated March 5, 2016

I can help you get started by joining you to a bunch of existing channels! Which types of channels would you like to join?

I'm interested in support

Developing on Mattermost

I don't know?

```
<div class="post-message post-message--collapsed" /></div>
<div class="attachment-actions">
  <button data-action-id="i7t33cbfa3rdbgb5cyicr-fxsia">I'm interested in support</button> == $0
  <button data-action-id="5g4cjrj9qazak9uyykezy">Developing on Mattermost</button>
  <button data-action-id="zs664ay46jfu5raqhzo7sb5saw">I don't know?</button>
</div>
```

Thank you

Mattermost is hiring!

<https://mattermost.com/careers>

Mattermost Mentor Program

- **Shepherding**
 - Helping someone get involved in the Mattermost community
 - Need help submitting your first pull request?
 - Want some general advice?
 - Two 1-1 sessions with a core contributor
- **Mentorship**
 - Focused on bringing more people into the Mattermost community
 - Application process
 - 3 month program
 - Minimum commitments must be met
 - Paired with a core contributor for the entire time
 - Weekly 1-1 sessions with a core contributor

Interested? Talk with me afterwards!

Questions?

How to reach me?

- Mattermost: @corey on <https://community.mattermost.com>
- Twitter: @corey_hulen
- Email: corey@hulen.com