

VIRTUAL CONFERENCES

How we used Drupal to save the company!



FLORIDA
DRUPAL CAMP



Conference
Catalysts



Goals & Takeaways

What do I get out of this?

- Real-world business examples
- Custom solutions and strategies
- How-To code examples
- Generate ideas for your own project
- *THIS IS A TECHNICAL TALK*

Who Are We

**Provider of Event &
Organizational
Management Solutions For
Professional Associations**

Conference Catalysts provides flexible solutions for professional associations to improve communication, organize efficiently, and run world-class events.



PLATINUM SPONSORS



Conference Catalysts provides flexible solutions for professional associations to improve communication, organize efficiently, and run world-class events.

Florida Drupal Camp 2022 Platinum Sponsor

Meet our Team

Digital Solutions Team

A tight-knit team of developers, designers, and technologists, the Digital Solutions Unit provides some of the most flexible, best-in-class digital experiences for associations and conferences.

**Developers, Graphic Artists,
and Content Specialists**





Who are we

Our Business; Our Clients

- IEEE Organizations
- Full Service On-Site Event Management
- Technical Paper Sessions



 **CONF LUX**
PLATFORM



What is CONFlux?

A virtual event web content delivery application.

Manage and organize Tracks, Sessions and their Presentations for access by Attendees.

Trusted by 100+ virtual conferences with 200,000+ combined attendees since April 2020.



Bringing CONFlux Online

Delivering a Product

- Data Consumption
- Deployments
- Migrations
- Design
- Execution/Delivery



Data Acquisition & Parsing Strategies

Data Consumption

- Variable Sources
- Google Sheets
- Custom Parsing
- Webform Submissions System
- Migration Preparation (API)

Data Acquisition & Parsing Strategies

Variable Sources

- Paper Management Systems
- No Consistency
- Interfacing with prepared data
- 3rd Party Registration systems

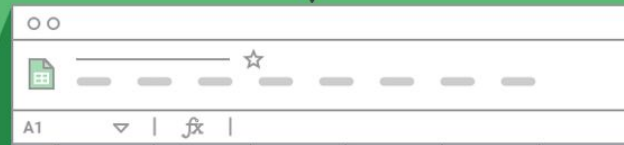
Data Acquisition & Parsing Strategies

Google Sheets

- Consistent and well known interface
- Boilerplate CSV for client interaction
- API, Scripting, Validation
- Sheetdb.io (REST API)

Collect and prepare data from client sources

spreadsheets.create



```
use League\Csv\Reader;
use League\Csv\CharsetConverter;
use League\Csv\Writer;
use PragmaRX\Countries\Package\Countries;

// Include helper functions
include_once 'inc/func.php';
include_once 'inc/google_api.php';

// Load Config File
$config = include('config.php');

// Google Sheets
$spreadsheetId = $config['spreadsheetId'];
$writeGoogleSheets = $config['writeGoogleSheets'];

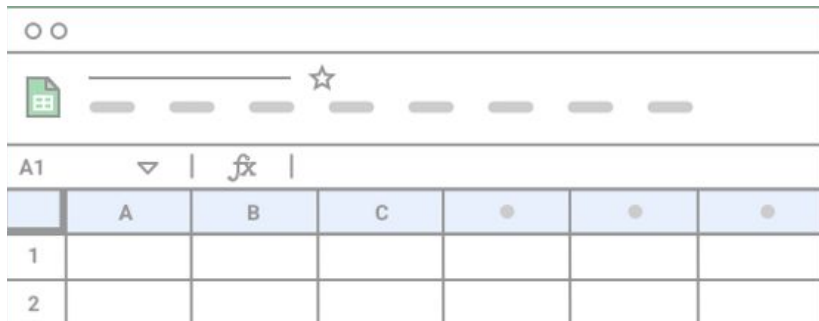
// Get the API client and construct the service object.
$client = getClient();
$service = new Google_Service_Sheets($client);
```

Data Acquisition & Parsing Strategies

Custom Parsing

- Flat File to Relational data
- Migration Preparation
- Unique Users, Sessions, Presentations, Tracks, etc

Massage data into a relational paradigm for Drupal Migrations



	A	B	C			
1						
2						

```
if($submissionsComplete) {
  // Submission Values
  $submissionValues = $service
  ->spreadsheets_values
  ->get($spreadsheetId, $googleSheetSubmissionRange)
  ->getValues();

  // Write master values to CSV
  $file = fopen($submissionsFile, 'w');
  foreach ($submissionValues as $line) {
    fputcsv($file, $line);
  }
  fclose($file);

  // load results
  /** @var \League\Csv\Reader */
  $submissions = Reader::createFromPath($submissionsFile, 'r');
  $submissions
  ->setHeaderOffset(0);
}

// Registrants
$reconcileRegistrants = $config['reconcileRegistrants'];
$registrantsRange = $config['registrantsRange'];

if($reconcileRegistrants) {
  $registrantValues = $service
  ->spreadsheets_values
  ->get($spreadsheetId, $registrantsRange)
  ->getValues();
  $registrantsFile = $config['registrantsFile'];

  // Write registrants values to CSV
  $file = fopen($registrantsFile, 'w');
  foreach ($registrantValues as $line) {
    fputcsv($file, $line);
  }
  fclose($file);

  // load results
  /** @var \League\Csv\Reader */
  $csvRegistrants = Reader::createFromPath($registrantsFile, 'r');
  $csvRegistrants->setHeaderOffset(0);
}
```



Data Acquisition & Parsing Strategies

Submission System

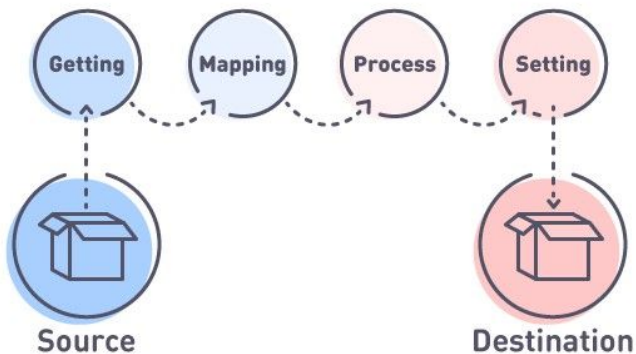
- Generated Prepopulated Links
- Mailmerge
- AWS File System
- `hook_..._options_alter`
- Export CVS

Data Acquisition & Parsing Strategies

Drupal Migrations

- Initiate database
- JSON Parser (Sheetdb.io)
- Migrate Update for Bulk
 - `overwrite_properties`

Parse data into a relational paradigm for Drupal Migrations



```
process:
  type:
    plugin: default_value
    default_value: user
  uid:
    -
      plugin: skip_on_empty
      method: row
      source: id
    -
      plugin: migration_lookup
      migration: cc_users
      source: id
  pass:
    plugin: callback
    callable: md5
    source: pw_plain
  destination:
    plugin: 'entity:user'
    md5_passwords: true
    overwrite_properties:
      - pass
```

```
$> drush mim --group=conflux
$> drush mim users --update
$> drush mim tracks,types,rooms
$> drush mim presentations --sync
```

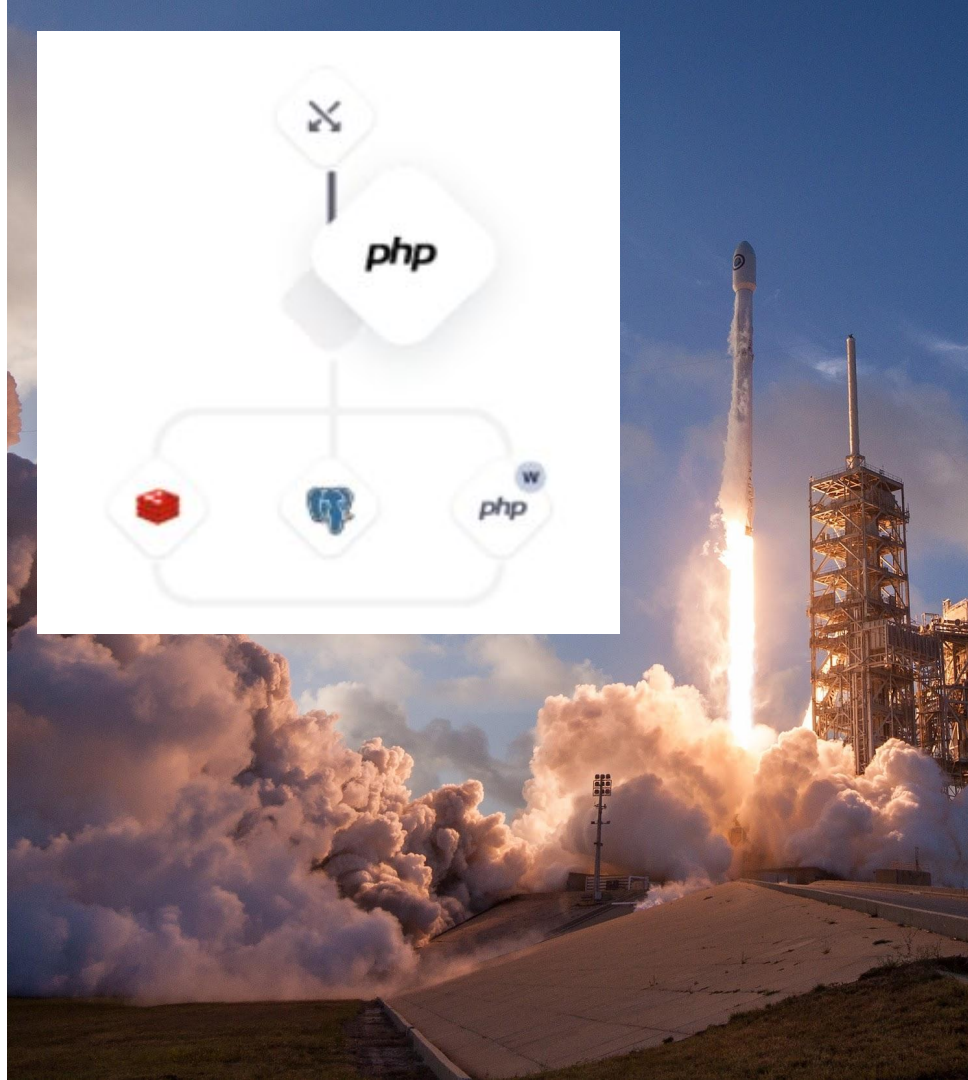
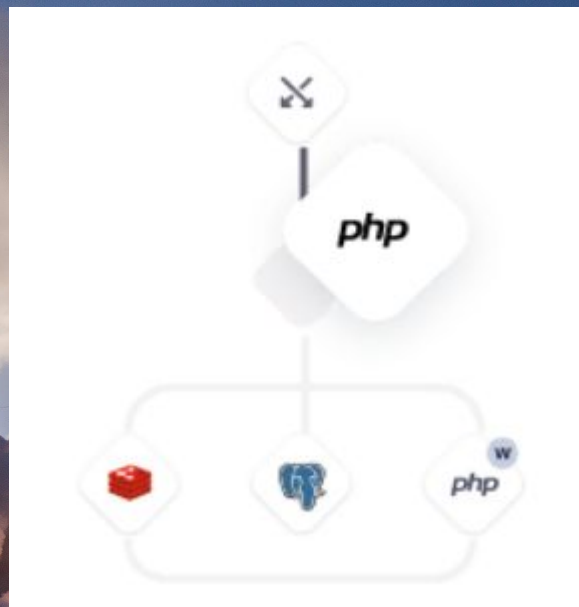
```
field_cc_date_range/value:
-
  plugin: skip_on_empty
  method: process
  source: start
-
  plugin: format_date
  from_format: 'n/j/Y G:i:s'
  to_format: 'Y-m-d\TH:i:s'
  source: start
```


Deployment

Platform.sh

- Scalable Resource Allocation
- Individualized deployments
- Multi-app (Node/Drupal)
- Containerized
- Fast. Reliable.

Manage your container
infrastructure with YAML and Git



Deployment

Drupal Installation

- drush si —existing-config
- Config Ignore
- Site Settings Configuration
- Stand-Alone Sites

Install Drupal via configuration
Think Installation Profile



Client Customizations

Design

- Customization directly in configuration
- Asset Injector (CSS/JS)
 - Shame
 - TW Variable Overrides
- View style templating
- TailwindCSS Custom Theme
- Custom Branding

Each application needs to be customized to meet client needs



Features

Drupal Features

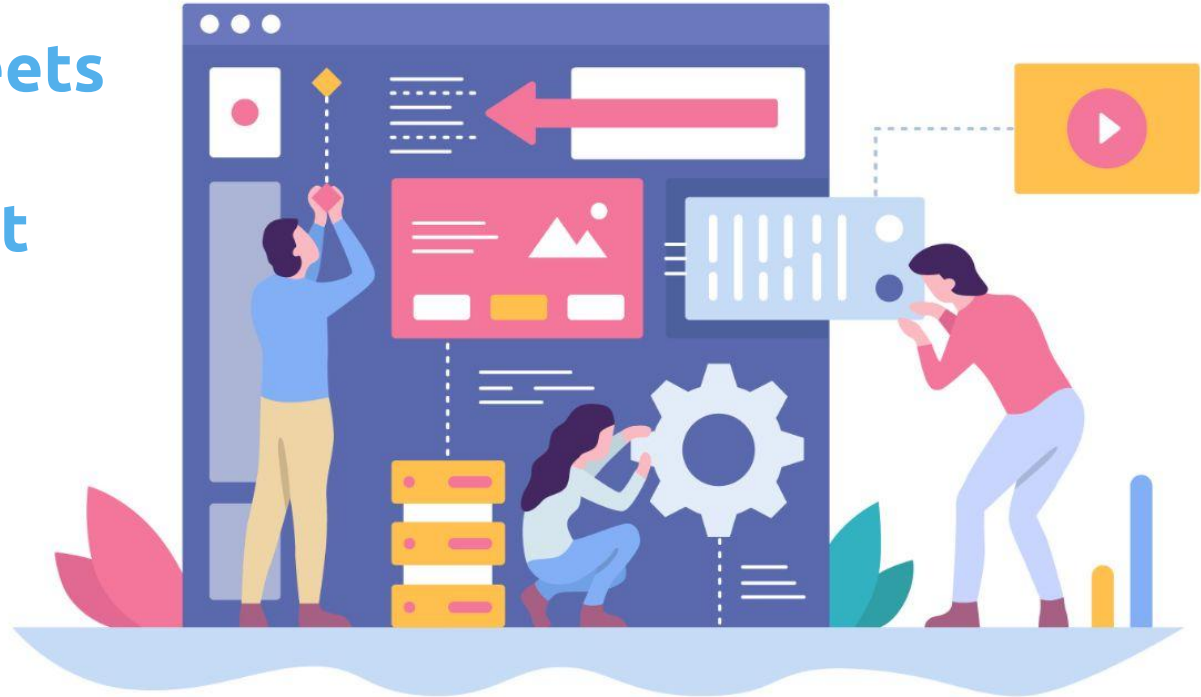
- Views Full Calendar
- Flag
- Site Settings and Labels
- Voting
- Webform

Drupal specific modules for rapid development and features



Integrations

- Google Sheets
- CVENT
- Rocket.Chat
- Jitsi





Integrations

Google Sheets API

- Bulk User Importing
- Assign Roles and Passwords
- On-demand (manual) import without duplication
- Drupal Batch API

The need:

- Been able to bulk import users from manually curated sheets in Google.
- Been able to assign passwords and roles via that Google sheet.
- Been able to import/update users on demand without duplicating existing ones.

Solution:

- Custom Drupal 9 Module

How the Module works:

- Fetches the rows in the Google sheet via JSON -> converts each user into a Batch API item.
- Creates a new user if it doesn't exist, updates the user if the user exist.
- Checks the roles and update the roles.

Caveats:

This process needs human intervention, triggered on demand.

Batch API:

Functions allowing forms processing to be spread out over several page requests, thus ensuring that the processing does not get interrupted because of a PHP timeout, while allowing the user to receive feedback on the progress of the ongoing operations.

Google Sheets Configuration Form

URL to data from.*

`https://docs.google.com/spreadsheets/d/[REDACTED]/gviz/t`

Has to be a JSON resource.

`https://docs.google.com/spreadsheets/d/SHEET_ID/gviz/tq?tx=out:json&sheet=User Imports`

Process Data

Save configuration



Updating 27 users

Running Batch "25" (User processed 24)



Completed 25 of 27.

93%



Integrations

CVENT API

- 3rd Party Registration
- Ultimate Cron
- Batch API
- Queue API

The need:

- Users registered in CVENT (<https://www.cvent.com/>) needed to be manually processed via Google Sheets, to assign password and roles.
- Once the conference started new users were created every minute, this made difficult to keep track of new users in Drupal vs new users in CVENT.
- Prevent missing data between imports.

Solution:

- Create a custom integration with CVENT via their API.
- Used the Queue API from Drupal.
- Used Drupal Cron jobs to fetch new users, and process users.
- Used Ultimate Cron to track imports and expose/manage Queue Workers

How the module works:

- Import of new users was splitted in two cron jobs:
 - **Cron 1:** Imports new users using `hook_cron()` based on a timestamp, keep track of the last import date and creates a queue item for later processing.
 - **Cron 2:** Queue Worker created that create users on Drupal mapping roles and even assigning passwords automatically.

Cron 1: is a **lean process**: and can fetch up to 200 users in each call, converts each user in JSON format to a queue item in the Drupal database.

Cron 2: is a **heavy process**: checks if a user exists in the database, if exists updated the roles of the user and the data that is new in CVENT, updates the password with the registration code given to the user.

Code example:

```
function cc_registrants_update_cron() {  
  
  $cvent_initialized = \Drupal::state()->get('cvent_initialized');  
  
  if (empty($cvent_initialized)) {  
    \Drupal::messenger()->addError('Configure the credentials for the API to work.');
```

```
    throw new Exception('Configure the CVENT API before executing this cron job.');
```

```
  }  
  
  $cvent_processor = \Drupal::service('cc_registrants_update.cvent_processing');  
  if ($cvent_processor->getTokenAndIterateAttendees()) {  
    $attendees = $cvent_processor->getAttendeesArray();  
  }  
  
  if (!empty($attendees)) {  
    $queue = \Drupal::service('queue');  
    $cvent_queue = $queue->get('cvent_queue');
```

```
    foreach ($attendees as $attendee) {  
      $attendee_object = $cvent_processor->createQueueItem($attendee);  
      if (!is_null($attendee_object)) {  
        $cvent_queue->createItem($attendee_object);  
      }  
    }  
  }  
}
```

Code example:

```
/**
 * Processes users from CVENT API.
 *
 * @QueueWorker(
 *   id = "cvent_queue",
 *   title = @Translation("CVENT Queue Worker"),
 *   cron = {"time" = 5}
 * )
 */
class CventQueueWorker extends QueueWorkerBase {

  /**
   * @param $data
   */
  public function processItem($data) {
    // ...
  }

  // ...
  protected function processUser($data) {
    // Check if user exists in the database.
    $user_from_drupal = user_load_by_mail($data->email);

    if (!$user_from_drupal) {
      $this->createUser($data);
    }
    else {
      $this->updateUser($data);
    }
  }
}
```

Customizations

User Merging

- Custom VueJS Interface
- Custom SQL queries to find effectively duplicates.
- User relationship matching
- Processively Decoupled Blocks

Retaining unique Users and merge all their content relationships and roles.



The need:

- Locate users that had the same First Name + Last Name combination.
- Avoid manual look-up and update of users and its content.
- Being able to mark **“no duplicate”** users

Solution:

- Created a custom pdb block **“Decoupled block”** with Vue.
- Created a SQL query that locates the duplicates using sub-queries
- Created a custom route that accepted two **UID** preventing to expose sensitive data to Vue.



Duplicates main app

Vue Merger

List of duplicated names:

- 21 - Brandon Ratzloff**
Username: BrandonRatzloff
brandon-razloff@example.com
Roles: Attendee, Affiliate
Presentations:
 - Panel: Going Faster - How to Cope with Shrinking Schedules and Increasing Complexity
 - UVVM: Bringing UVM to VHDL**Affiliation:** Confcats
[Not a duplicate](#)
- 23 - Brandon Ratzloff**
Username: BrandonRatzloff12
brandon@conferencecatalysts.com
Roles: Attendee, Affiliate
Affiliation: Conference Catalyst
[Not a duplicate](#)
- 197 - Israel Morales**
Username: Israel-Morales-84
israel.morales@example.com
Roles: Author
Presentations:
 - Panel: Going Faster - How to Cope with Shrinking Schedules and Increasing Complexity**Affiliation:** Conference Catalyst
[Not a duplicate](#)
- 600 - Israel Morales**
Username: IsraelMorales
israel.mo@confcats.io

tps://conflux.indo.site/user/21

Merge From

Merge To



Merge

Vue Merger Demo:

The routes:

```
cc_duplicates.merger_request:
  path: '/admin/duplicates/merge_users/{mergeFrom}/{mergeTo}'
  defaults:
    _controller: '\Drupal\cc_duplicates\Controller\DuplicatesController::process'
  methods: ['POST']
  requirements:
    _permission: 'administer site configuration'
    _format: json

cc_duplicates.no_duplicate:
  path: '/admin/duplicates/no_duplicate/{id}'
  defaults:
    _controller: '\Drupal\cc_duplicates\Controller\DuplicatesController::noDuplicate'
```

Merge method example:

```
private function mergeUsers(User $merge_from_user, User $merge_to_user): bool {  
  
    // Presentations:  
    $this->updatePresentations($merge_from_user, $merge_to_user);  
  
    // Media:  
    $this->updateMedia($merge_from_user, $merge_to_user);  
  
    // Biography  
    $this->mergeFieldItemList('field_cc_affiliation',  
        $merge_from_user, $merge_to_user);  
  
    // Affiliation  
    $this->mergeFieldItemList('field_cc_bio',  
        $merge_from_user, $merge_to_user);  
  
    // Roles:  
    $this->mergeRoles($merge_from_user, $merge_to_user);  
  
    // Set alternative email from merge from.  
    $this->addAlternativeEmail($merge_from_user, $merge_to_user);  
  
    // Delete duplicated user.  
    $this->deleteUser($merge_from_user);  
  
    $merge_to_user->save();  
  
    return TRUE;  
}
```

Integrations

Engage (Jitsi + Rocket.Chat)

- Networking, Affiliate Booths, Breakouts
- One on One Video chat
- Jitsi + Rocket.Chat + Custom Code
- Live Example



CONflux Admin...
⋮ 🗙 ⭐




Start of conversation

😊 Message

CONFLUX ENGAGEMENT // 📍 NETWORKING

Test

 **CHECK IN!**
Click here to check in and gain CONFlux Credits!

INVITE OTHERS

Invite Attendees

 **Invite Attendees**

MODERATOR



Integrations

Jitsi

- Jitsi Meet video conferencing platform
- Jitsi as a Service (JaaS)
- IFrame API
- Channel Field

Thumbnail for participant 'CN' with mute and video off icons.



Thumbnail for participant 'CONflux Admin' with mute, video off, and star icons.

CONflux Admin

Bottom toolbar containing icons for mute, video, chat, help, app grid, participant 'CN', HD video, full screen, and a menu icon.



Jitsi

Jitsi Servers vs Jitsi Services

- One Click Application
 - Digital Ocean, Linode
- Build from Source
- JaaS (jaas.8x8.vc)

Self Hosted video conferencing



Jitsi

Jitsi IFrame API

- Libraries
- Drupal.behaviors

```
# Libraries
engage:
  js:
    https://engage.example.io/external_api.js: { type: external }
```

```
// Jitsi IFrame API
(function($, Drupal) {
  'use strict';
  Drupal.behaviors.engage = {
    attach: function(context, settings) {
      $(document, context).once('engage').each( function() {
        ...
        // Room Options
        const options = {
          roomName: drupalSettings.engage.roomName,
          width: "100%",
          height: 600,
          parentNode: document.querySelector('#meet'),
          configOverwrite: configOverwriteSetting,
          interfaceConfigOverwrite: interfaceConfigOverwriteSetting,
          userInfo: {
            email: drupalSettings.user.mail,
            displayName: drupalSettings.user.displayName
          },
        },
      );
      // Create Room
      const api = new JitsiMeetExternalAPI(domain, options);
      ...
    }
  };
});
```

Integrations

Rocket.Chat

- Embedded chat/Omnichannel for online support
- Custom module using its REST API
- Deploy strategy using platform.sh
- Open Source Slack Alternative

Unread

important 2

Conversations

- # product-releases
- # new-employees
- Brett Smith
- Ann Miller
- # marketing-agency
- # contact center



Marc Davis 9:09 AM

03:11

product-releases

Mark Collins 9:09 AM
Hi! @all could you please take a look at this

Product release September 2021

2 2

Daniel Smith 9:09 AM
On it!

Reply 1 Yesterday at 10:41 PM

Amber Garzia 9:09 AM
@Marc Collins I think we need more con

Adding to Jira!

2

Jira 9:10 AM

Product Release: Integrations
Status: done
Priority: high

Your message...

product-releases

Your message...

Rocket.Chat

Custom module

- Using [HTTP Client Manager](#) - Service Descriptions via YAML
- Single-sign on for embedded channel
- User updates

```
/**
 * Gets Rocket Chat auth token.
 *
 * @param \Symfony\Component\HttpFoundation\Request $request
 * - The request.
 *
 * @return \Symfony\Component\HttpFoundation\Response
 * - A response which contains the ID and CSRF token.
 */
public function handle(Request $request) {
    $format = $this->getRequestFormat($request);
    if (!$this->currentUser()->getAccount()->isAnonymous()) {
        $auth_token = $this->httpClient->rocketchatUserLogin($this->currentUser()->getAccount());
        if (empty($auth_token)) {
            $response = [];
            $response['loginToken'] = $auth_token['data']['authToken'];
            $encoded_response_data = $this->serializer->encode($response, $format);
            return new Response($encoded_response_data);
        }
    }

    return new Response('', Response::HTTP_UNAUTHORIZED);
}
```

```
operations:
  userCreate:
    httpMethod: "POST"
    uri: "users.create"
    summary: "Create a new user. Re..."
    parameters:
      X-Rocketchat-EC-Auth:
        description: "This request
        default: true
        static: true
        location: header
      email:
        location: "json"
        description: "The email add
        type: "string"
        required: true
      name:
        location: "json"
        description: "The display name of the user."
        type: "string"
        required: true
      password:
        location: "json"
        description: "The password for the user."
        type: "string"
        required: true
      username:
        location: "json"
        description: "The username for the user."
        type: "string"
        required: true
    responseClass: GeneralObject

public function userCreate(AccountInterface $account) {
    $user_id = NULL;
    $config = $this->configFactory->get('rocketchat_ec.settings');
    $email = $account->getEmail();

    $cache_key = 'user:' . $account->id();
    $cache = $this->cache->get($cache_key);
    if ($cache) {
        $user_id = $cache->data;
    }
    else {
        // Find user by email address
        $query = '{"emails": {"$elemMatch": {"address": {"$eq": "' . $email . '"}}}}';
        $response = $this->userList(['query' => $query]);
        if (empty($response['count'])) {
            $user_id = $response['users'][0]['_id'];
        }
        else {
            $rcuser = $this->userCreate([
                'email' => $email,
                'name' => $this->rocketchatDisplayName($account),
                // Generate a random password, as we are using auth tokens
                'password' => user_password(64),
                'username' => $this->rocketchatUsername($account),
            ]);
            $user_id = $rcuser['user']['_id'];
        }
        if (empty($user_id)) {
            $this->cache->set($cache_key, $user_id);
        }
    }
    return $user_id;
}
```


Rocket.Chat

Custom module

- RC [iFrame Integration](#)
- Single sign-on

Accounts

Save changes

Iframe

Enabled

Iframe URL

https://conflux.Indo.site:57440/

API URL

https://conflux.Indo.site:57440/rocketchat_ec/token

Api Method

POST

Reset Section to Default

```
rocketchat_ec.token:  
  - path: '/rocketchat_ec/token'  
  - defaults:  
    - _controller: \Drupal\rocketchat_ec\Controller\AuthToken::handle  
    - methods: [POST]  
    - requirements:  
      - _permission: 'request auth token'  
      - _format: 'json'
```

```
public function handle(Request $request) {  
    $format = $this->getRequestFormat($request);  
    if (!$this->currentUser()->getAccount()->isAnonymous()) {  
        $auth_token = $this->httpClient->_rocketchatUserLogin($this->currentUser()->getAccount());  
        if (!empty($auth_token)) {  
            $response = [];  
            $response['loginToken'] = $auth_token['data']['authToken'];  
            $encoded_response_data = $this->serializer->encode($response, $format);  
            return new Response($encoded_response_data);  
        }  
    }  
    return new Response('', Response::HTTP_UNAUTHORIZED);  
}
```

Rocket.Chat

Platform.sh

- Multi-app
- Really flexible, installing mongo and RC manually on a node container.
- Multi-environment
- Auto-initialization through environment variables, shell and python scripts.

```
##
# Initialize replica set
#
if [ ! -f $PLATFORM_APP_DIR/run/mongodb/initialized ]; then
  echo "Initializing mongodb..."
  for i in `seq 1 60`; do
    nc -z localhost 27017 &&
    mongo --eval "res = rs.initiate(); assert.commandWorked(res);" &&
    touch $PLATFORM_APP_DIR/run/mongodb/initialized &&
    pm2 restart rocket.chat &&
    break
  done
  echo "Retry $i/60 initializing mongodb. Waiting 2 sec.."
  sleep 2
done
fi

##
# Admin password change
#
if [ -n "$ROCKETCHAT_ADMIN_PASSWORD" ]; then
  PASSWORD_LENGTH=$((#${ROCKETCHAT_ADMIN_PASSWORD} - 2))
  PASSWORD_ASTERISKS= `printf "%${PASSWORD_LENGTH}s" |tr " " "*"`
  MASKPASSWORD="${ROCKETCHAT_ADMIN_PASSWORD::1}${PASSWORD_ASTERISKS}${ROCKETCHAT_ADMIN_PASSWORD: -1}"
  PASSWORD=`python3 $PLATFORM_APP_DIR/scripts/adminpw.py`
  echo "Updating rocketchat admin password to $MASKPASSWORD through hash $PASSWORD..."
  for i in `seq 1 120`; do
    mongo rocketchat --eval "res = db.getCollection('users').update({username:'$ADMIN_USERNAME'}, { \$set: {'services':
    break
  done
  echo "Retry $i/120 changing admin pw. Waiting 2 sec.."
  sleep 2
done
fi
```

```
OVERWRITE_SETTING_Site_Name="CONFlux Engage"
OVERWRITE_SETTING_Server_Type="Private Team"
OVERWRITE_SETTING_Register_Server=false
```

```
OVERWRITE_SETTING_Accounts_AllowEmailNotifications=false
OVERWRITE_SETTING_Accounts_ShowFormLogin=false
```

```
OVERWRITE_SETTING_Accounts_iframe_enabled=true
OVERWRITE_SETTING_Accounts_TwoFactorAuthentication_By_Email_Auto_Opt_In=false
OVERWRITE_SETTING_Accounts_TwoFactorAuthentication_Enforce_Password_Fallback=false
```

Rocket.Chat

Platform.sh

- Some RC settings cannot be updated through ENV VARS

```
##
# Rocket.chat
#
mkdir -p /tmp/rocketchat
curl -L https://releases.rocket.chat/$ROCKETCHAT_VERSION/download -o /tmp/rocketc
tar -xzf /tmp/rocketchat/rocket.chat.tgz -C /tmp/rocketchat
cd /tmp/rocketchat/bundle/programs/server
npm install
mv /tmp/rocketchat/bundle $PLATFORM_APP_DIR/rocket.chat
cd -
```

```
##
# MongoDB - We install it local instead of using a service due to missing
# support for replication sets.
```

```
#
mkdir -p /tmp/mongodb
curl -L https://fastdl.mongodb.org/linux/mongodb-linux-x86_64-debian92-$MONGODB_VERSION.tgz -o /tmp/mongodb/mongodb.tgz
tar -xzf /tmp/mongodb/mongodb.tgz -C /tmp/mongodb --strip 1
mv /tmp/mongodb/bin $PLATFORM_APP_DIR/mongodb
```

```
..# Complex settings that cannot be updated with OVERWRITE_SETTING_ENV vars
..settings = {
..  ..'Hide_System_Messages': {
..  ..  ..'value': ['au'],
..  ..  },
..  },
..}

..for key, value in settings.items():
..  ..r = requests.post(api_url + '/settings/' + key, json=value, headers=headers)
..  ..if not r.json()['success']:
..  ..  ..errno = 2
..  ..  ..print("Updating", key + "...", r.json()['success'])

..# Permissions update
..roles_non_user = ["admin", "bot", "app"]
..perms = {
..  .."permissions": [
..  ..  ..{
..  ..  ..  .."_id": "create-c",
..  ..  ..  .."roles": roles_non_user
..  ..  ..},
..  ..  ..{
..  ..  ..  .."_id": "create-d",
..  ..  ..  .."roles": roles_non_user
..  ..  ..},
..  ..  ]
..  }
..  r = requests.post(api_url + '/permissions.update', json=perms, headers=headers)
..  print("Updating permissions...", r.json()['success'])
```



 **CONF LUX**
PLATFORM





Analyzing the Results

Statistics

- Over 25,000 Presentation Submissions
- 96 Conference deployments
- Over 66,000 of Attendees
- 1000s of Zoom calls

Job Opportunities

We're Hiring!

Join a team where you can have an immediate impact.

- Drupal + Symfony
- Vue.js + Inertia.js + Laravel + Tailwind CSS
- Intermediate to Senior Positions available

Welcome developers, designers, and creators!



**Conference
Catalysts**

Now Hiring

Senior Full-Stack Developer

**Apply at the
Conference
Catalysts Booth**