Heart Screening Management System – Milestone 2

Team Members
Carlo Campanini
Chris Newberry
Drew Dunkelberger
John Dewey
Noah Wilson

Client
Evan Ernst, CEO
Klynton Holmes, Tech Advisor

Faculty Sponsor
Dr. Eraldo Ribeiro

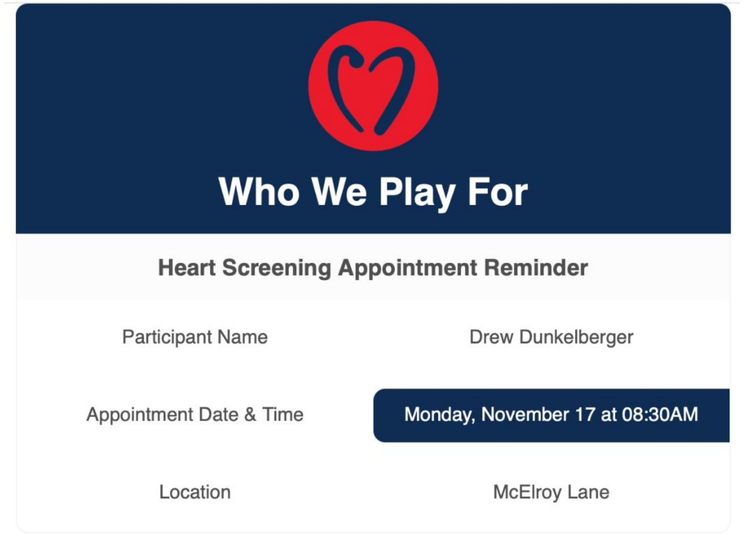
Milestone 2 Overview

Implementing, test, and demo features:

- Email reminder notifications day before event
- Close event registration day before event
- Vary verbiage depending on participant’s age during registration
- Auto display available times slots for private events
- Provide unique registration link for event directors to distribute to participants

Automated Event Email Reminders Content

- Rendered HTML:

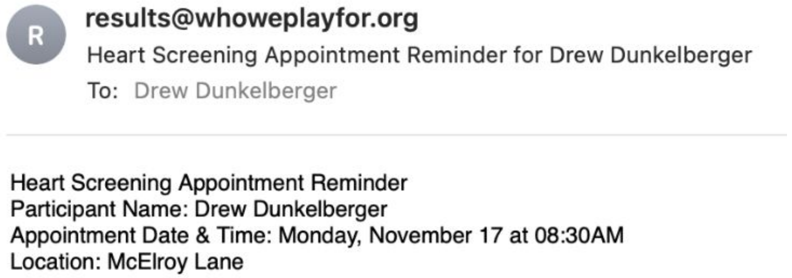


Who We Play For

Heart Screening Appointment Reminder

Participant Name	Drew Dunkelberger
Appointment Date & Time	Monday, November 17 at 08:30AM
Location	McElroy Lane

- Simple text alternative, when HTML is not supported for mail client:

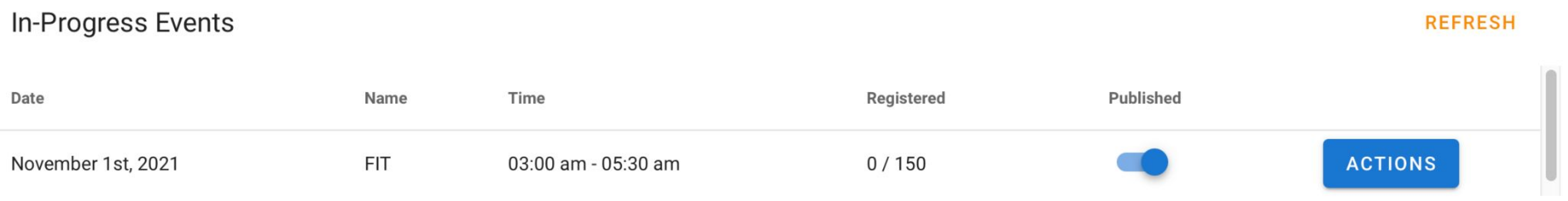


results@whoweplayfor.org
Heart Screening Appointment Reminder for Drew Dunkelberger
To: Drew Dunkelberger

Heart Screening Appointment Reminder
Participant Name: Drew Dunkelberger
Appointment Date & Time: Monday, November 17 at 08:30AM
Location: McElroy Lane

Automated Event Finalization Director UI

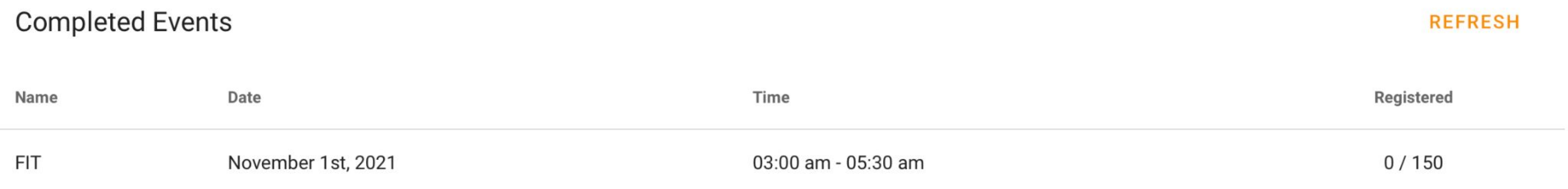
- In progress event:



In-Progress Events

Date	Name	Time	Registered	Published	
November 1st, 2021	FIT	03:00 am - 05:30 am	0 / 150		ACTIONS

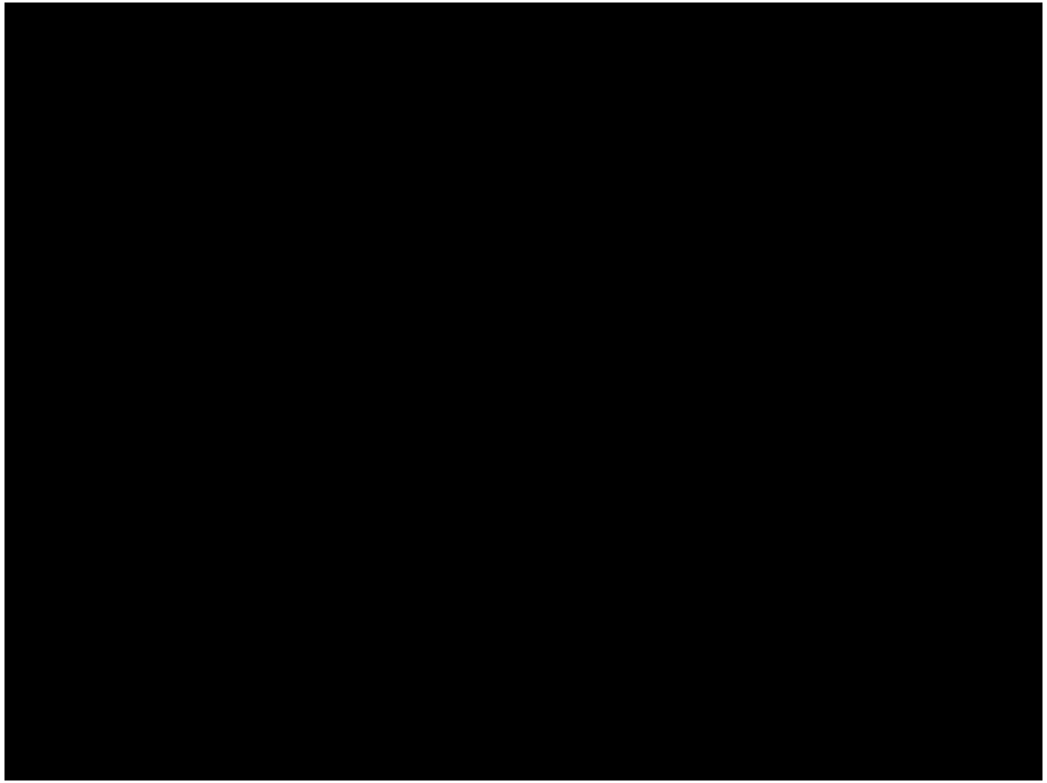
- Finalized event:



Completed Events

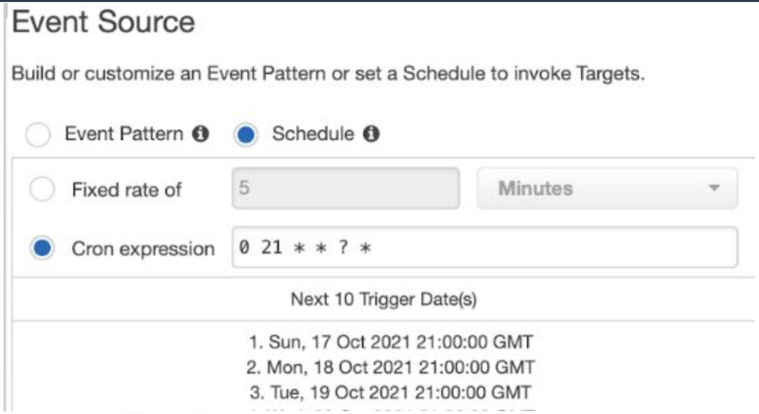
Name	Date	Time	Registered
FIT	November 1st, 2021	03:00 am - 05:30 am	0 / 150

Event Reminder/Finalization Demo



Event Reminder/Finalization Implementation

- AWS EventBridge Rule for cron job:



Event Source

Build or customize an Event Pattern or set a Schedule to invoke Targets.

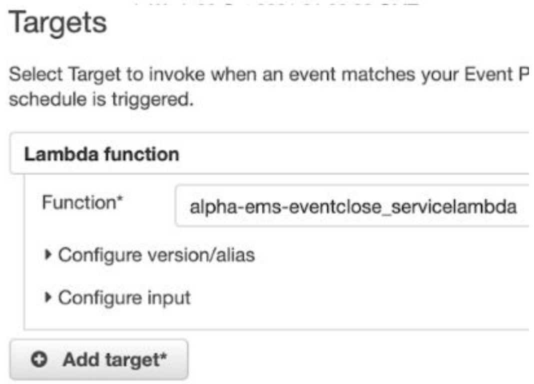
Event Pattern [✓ Schedule]

Fixed rate of 5 Minutes

[✓ Cron expression] `0 21 * * ? *`

Next 10 Trigger Date(s)

1. Sun, 17 Oct 2021 21:00:00 GMT
2. Mon, 18 Oct 2021 21:00:00 GMT
3. Tue, 19 Oct 2021 21:00:00 GMT



Targets

Select Target to invoke when an event matches your Event P
schedule is triggered.

Lambda function

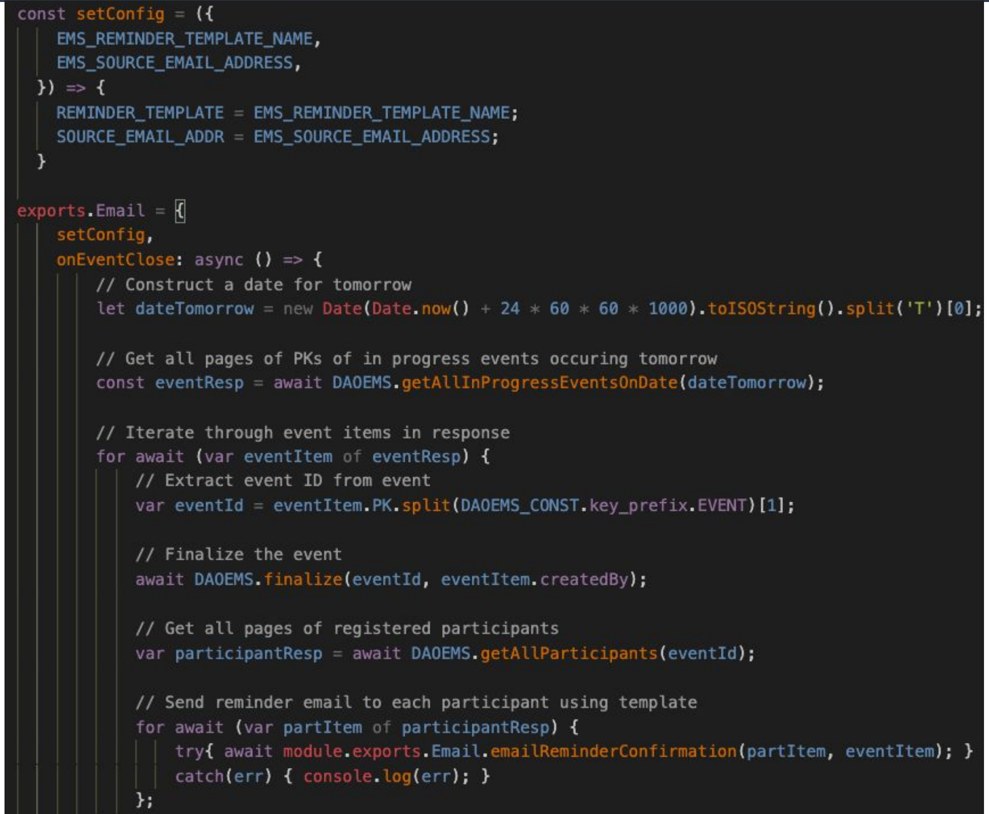
Function*: alpha-ems-eventclose_servicelambda

▸ Configure version/alias

▸ Configure input

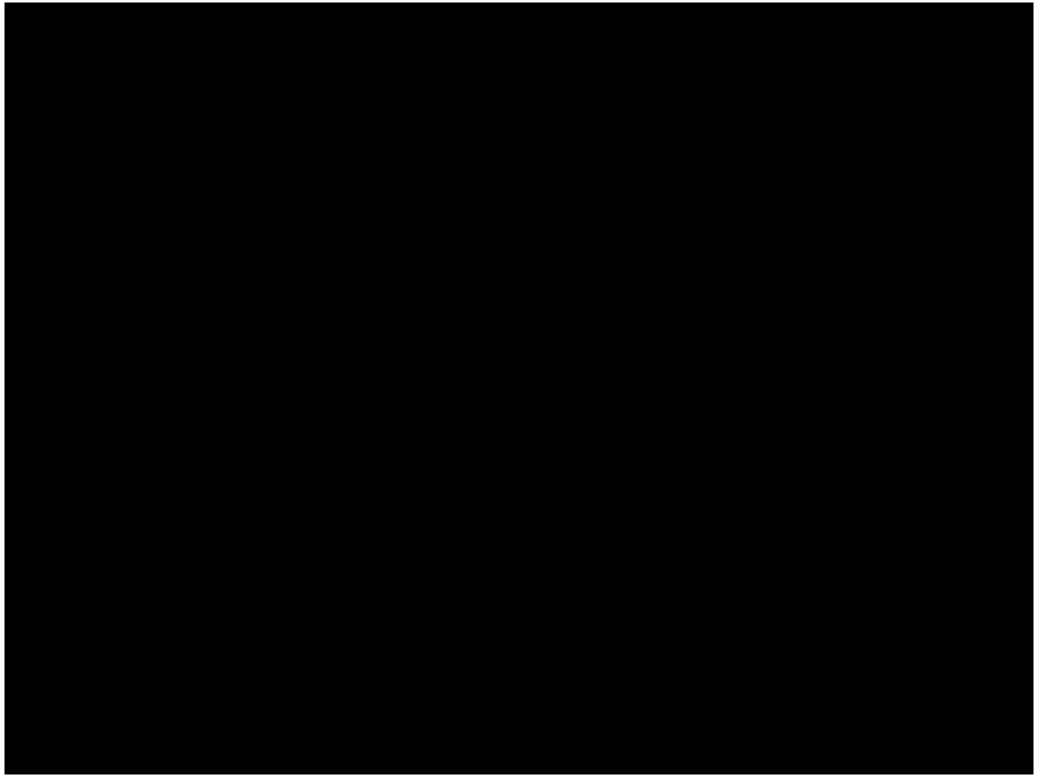
Add target*

- Lambda Function:



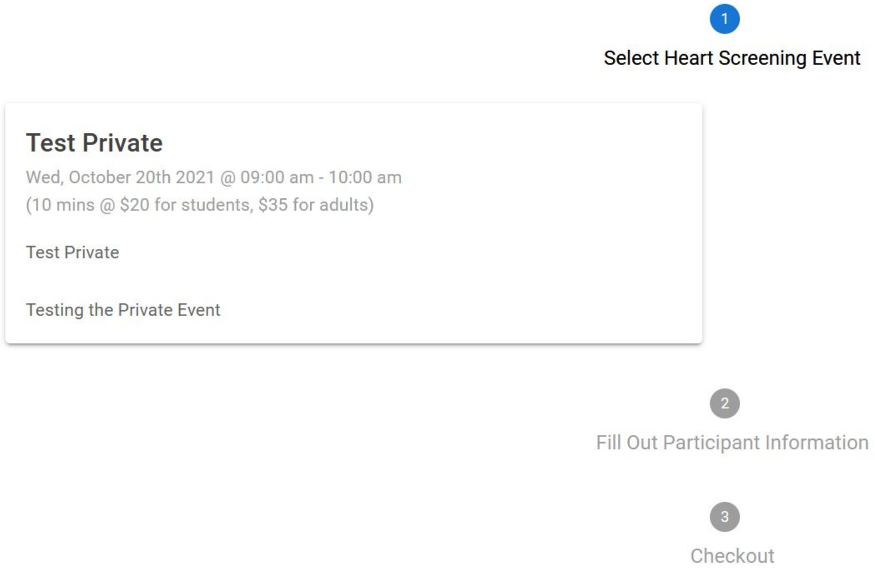
```
const setConfig = ({
    EMS_REMINDER_TEMPLATE_NAME,
    EMS_SOURCE_EMAIL_ADDRESS,
  }) => {
    REMINDER_TEMPLATE = EMS_REMINDER_TEMPLATE_NAME;
    SOURCE_EMAIL_ADDR = EMS_SOURCE_EMAIL_ADDRESS;
  }

exports.Email = {
    setConfig,
    onEventClose: async () => {
        // Construct a date for tomorrow
        let dateTomorrow = new Date(Date.now() + 24 * 60 * 60 * 1000).toISOString().split('T')[0];

        // Get all pages of PKs of in progress events occuring tomorrow
        const eventResp = await DAOEMS.getAllInProgressEventsOnDate(dateTomorrow);

        // Iterate through event items in response
        for await (var eventItem of eventResp) {
            // Extract event ID from event
            var eventId = eventItem.PK.split(DAOEMS_CONST.key_prefix.EVENT)[1];

            // Finalize the event
            await DAOEMS.finalize(eventId, eventItem.createdBy);

            // Get all pages of registered participants
            var participantResp = await DAOEMS.getAllParticipants(eventId);

            // Send reminder email to each participant using template
            for await (var partItem of participantResp) {
                try{ await module.exports.Email.emailReminderConfirmation(partItem, eventItem); }
                catch(err) { console.log(err); }
            };
```

Age-Dependent Verbiage Demo



Auto-display Private Event Time Slots

- Before implementing auto-display:



1 Select Heart Screening Event

Test Private

Wed, October 20th 2021 @ 09:00 am - 10:00 am
(10 mins @ $20 for students, $35 for adults)

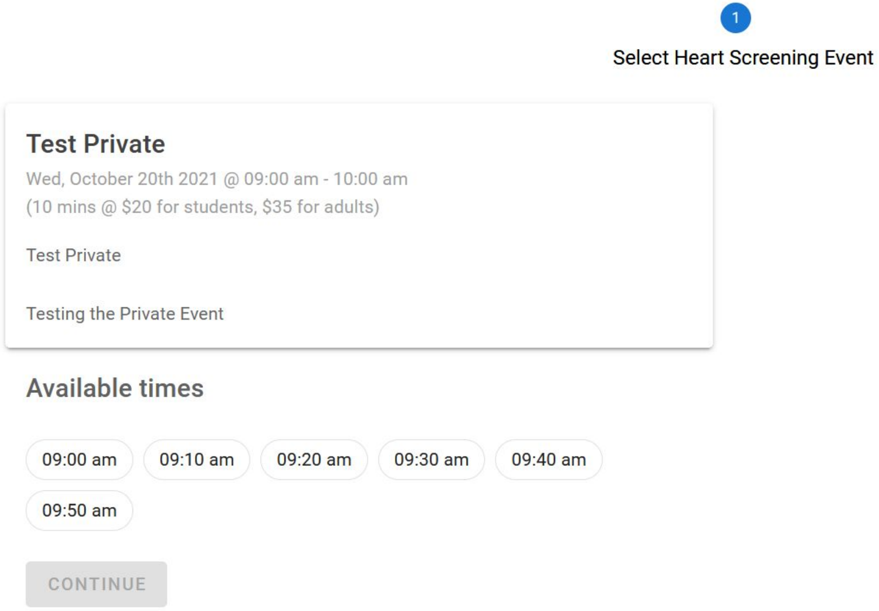
Test Private

Testing the Private Event

2 Fill Out Participant Information

3 Checkout

- After implementing auto-display:



1 Select Heart Screening Event

Test Private

Wed, October 20th 2021 @ 09:00 am - 10:00 am
(10 mins @ $20 for students, $35 for adults)

Test Private

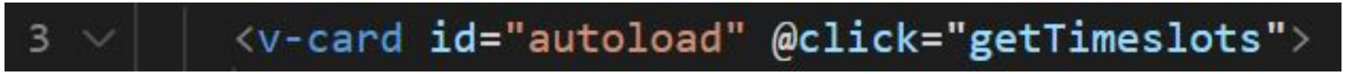
Testing the Private Event

Available times

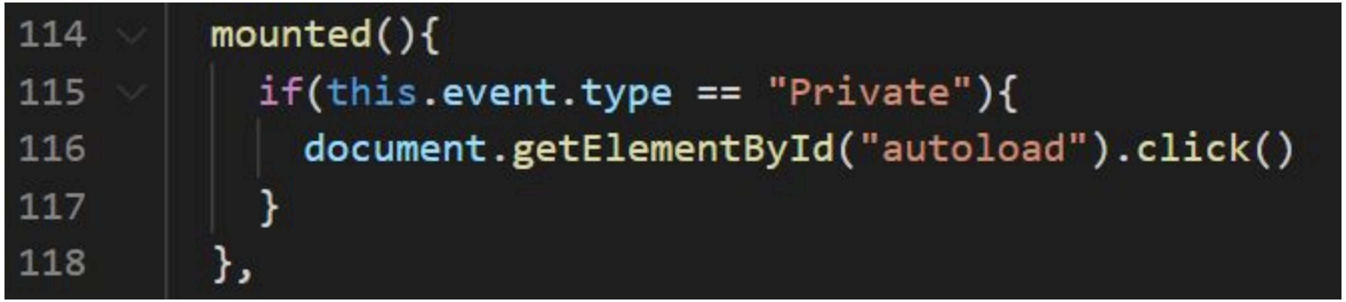
09:00 am | 09:10 am | 09:20 am | 09:30 am | 09:40 am | 09:50 am

CONTINUE

Auto-Display Private Event Time Slots



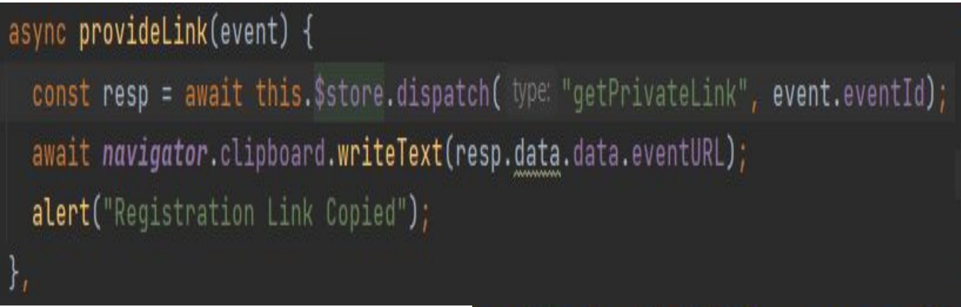
```
  <v-card id="autoload" @click="getTimeslots">
```



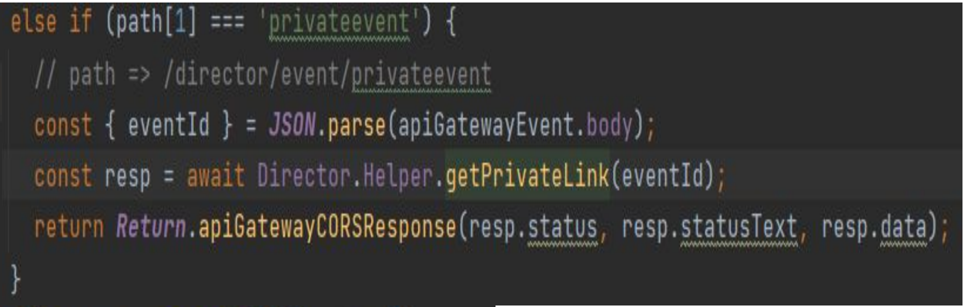
```
mounted(){
  if(this.event.type == "Private"){
    document.getElementById("autoload").click()
  }
},
```

Event Link Functionality Demo

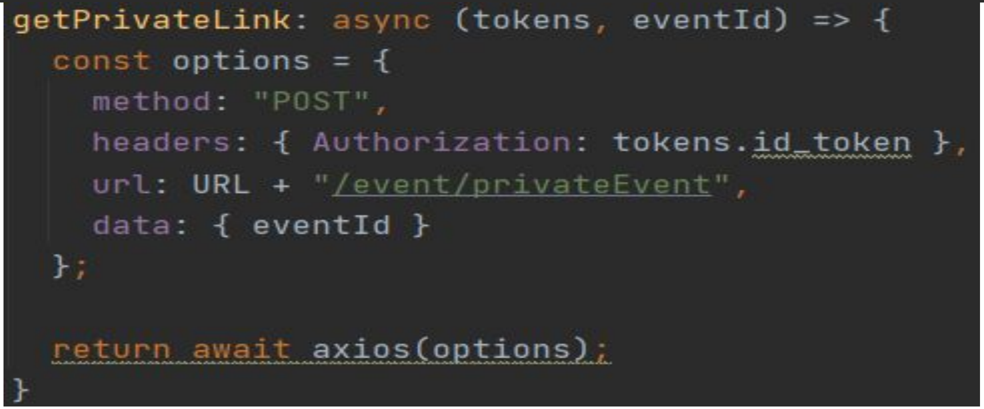
Event Link Functionality Implementation



```
async provideLink(event) {
  const resp = await this.$store.dispatch( type: "getPrivateLink", event.eventId);
  await navigator.clipboard.writeText(resp.data.data.eventURL);
  alert("Registration Link Copied");
},
```



```
else if (path[1] === 'privateevent') {
  // path => /director/event/privateevent
  const { eventId } = JSON.parse(apiGatewayEvent.body);
  const resp = await Director.Helper.getPrivateLink(eventId);
  return Return.apiGatewayCORSResponse(resp.status, resp.statusText, resp.data);
}
```



```
getPrivateLink: async (tokens, eventId) => {
  const options = {
    method: "POST",
    headers: { Authorization: tokens.id_token },
    url: URL + "/event/privateEvent",
    data: { eventId }
  };

  return await axios(options);
}
```

Milestone 3 Tasks
(in order of priority)

Research and test feasibility:

1. Dedicated LAN machine for on-site registration
    a. Machine will need to interact with our database to update as participants register **on-site**

Implement, test, and demo:

2. Search payments by several filters
    a. Date, date range, participant, event
3. Export payments to spreadsheet

Begin implementation:

4. Participant cancel registration
    a. Includes refunding functionality

Questions?