master

greenlight / app / assets / javascripts / room.js / <> Jump to

hadicheaito1 Preventing XSS Share Room Access (#3279) … ✓   History

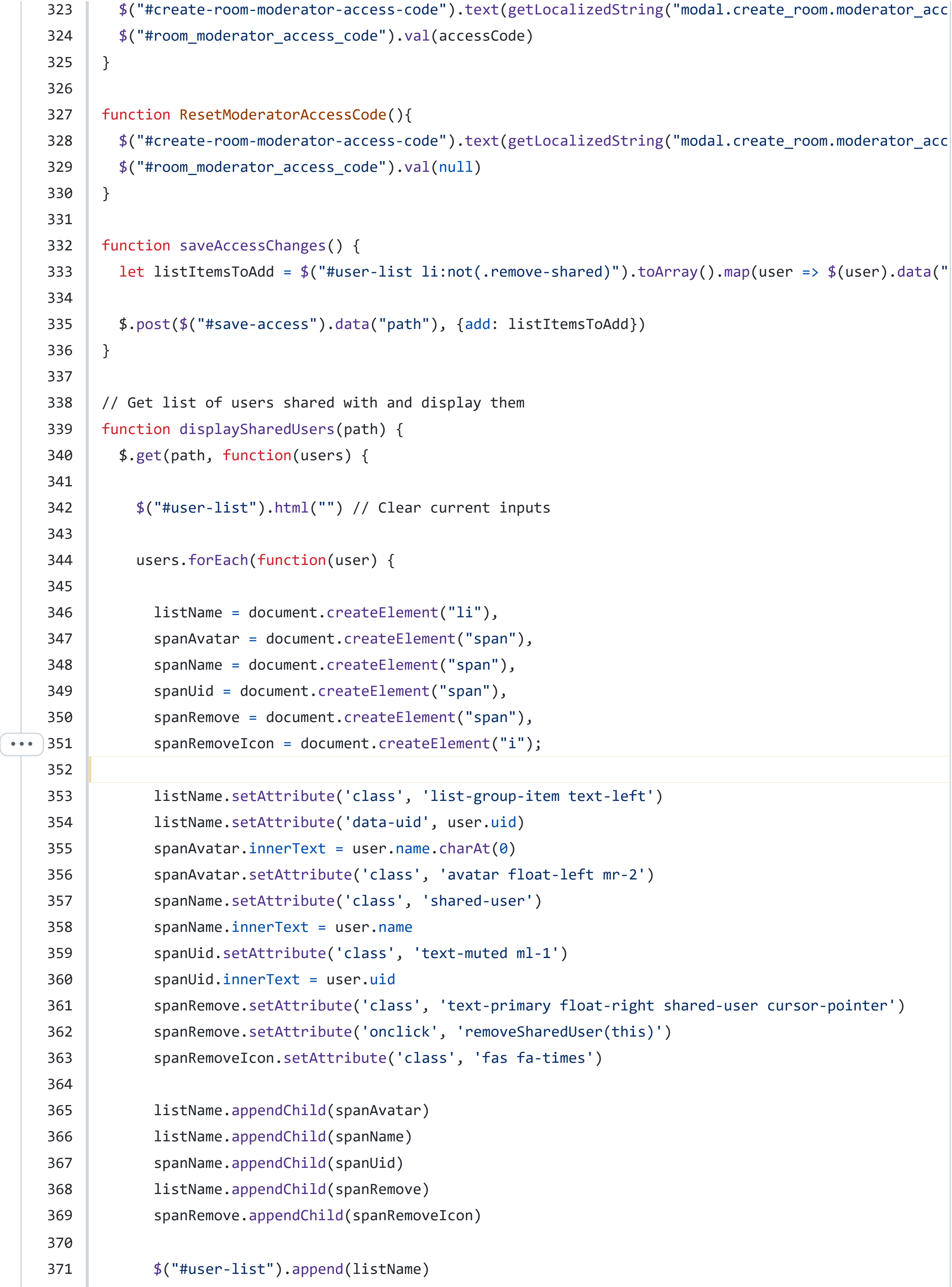
26 contributors +14

517 lines (433 sloc) | 18.8 KB

```javascript
// BigBlueButton open source conferencing system - http://www.bigbluebutton.org/.
//
// Copyright (c) 2018 BigBlueButton Inc. and by respective authors (see below).
//
// This program is free software; you can redistribute it and/or modify it under the
// terms of the GNU Lesser General Public License as published by the Free Software
// Foundation; either version 3.0 of the License, or (at your option) any later
// version.
//
// BigBlueButton is distributed in the hope that it will be useful, but WITHOUT ANY
// WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A
// PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.
//
// You should have received a copy of the GNU Lesser General Public License along
// with BigBlueButton; if not, see <http://www.gnu.org/licenses/>.

// Room specific js for copy button and email link.
$(document).on('turbolinks:load', function(){
  var controller = $("body").data('controller');
  var action = $("body").data('action');

  // highlight current room
  $('.room-block').removeClass('current');
  $('a[href="' + window.location.pathname + '"] .room-block').addClass('current');

  // Only run on room pages.

  if (controller == "rooms" && action == "show"){
    // Display and update all fields related to creating a room in the createRoomModal
    $("#create-room-block").click(function(){
      showCreateRoom(this)
    })

    checkIfAutoJoin()
  }

    // Autofocus on the Room Name label when creating a room only
  $('#createRoomModal').on('shown.bs.modal', function (){
    if ($(".create-only").css("display") == "block"){
      $('#create-room-name').focus()
    }
  })

  if (controller == "rooms" && action == "show" || controller == "admins" && action == "server_roo
    // Display and update all fields related to creating a room in the createRoomModal
    $(".update-room").click(function(){
      showUpdateRoom(this)
    })

    // share room pop up accessibility
    manageAccessAccessibility();

    $(".delete-room").click(function() {
      showDeleteRoom(this)
    })

    // For keyboard users to be able to generate access code
    generateAccessCodeAccessibility()

    $('.selectpicker').selectpicker({
      liveSearchPlaceholder: getLocalizedString('javascript.search.start')
    });
    // Fixes turbolinks issue with bootstrap select
    $(window).trigger('load.bs.select.data-api');

    $(".share-room").click(function() {
      // Update the path of save button
      $("#save-access").attr("data-path", $(this).data("path"))
      $("#room-owner-uid").val($(this).data("owner"))

      // Get list of users shared with and display them
      displaySharedUsers($(this).data("users-path"))
    })

    $("#shareRoomModal").on("show.bs.modal", function() {
      $(".selectpicker").selectpicker('val','')
    })

    $(".bootstrap-select").on("click", function() {
      $(".bs-searchbox").siblings().hide()
    })

    $("#share-room-select ~ button").on("click", function() {
      $(".bs-searchbox").siblings().hide()
    })

    $(".bs-searchbox input").on("input", function() {
      if ($(".bs-searchbox input").val() == '' || $(".bs-searchbox input").val().length < 3) {
        $(".select-options").remove()
        $(".bs-searchbox").siblings().hide()
      } else {
        // Manually populate the dropdown
        $.get($("#share-room-select").data("path"), { search: $(".bs-searchbox input").val(), owne
          $(".select-options").remove()
          if (users.length > 0) {
            users.forEach(function(user) {
              let opt = document.createElement("option")
              $(opt).val(user.uid)
              $(opt).text(user.name)
              $(opt).addClass("select-options")
              $(opt).attr("data-subtext", user.uid)
              $("#share-room-select").append(opt)
            })
            // Only refresh the select dropdown if there are results to show
            $('#share-room-select').selectpicker('refresh');
          }
          $(".bs-searchbox").siblings().show()
        })
      }
    })

    $(".remove-share-room").click(function() {
      $("#remove-shared-confirm").parent().attr("action", $(this).data("path"))
    })

    // User selects an option from the Room Access dropdown
    $(".bootstrap-select").on("changed.bs.select", function(){
      // Get the uid of the selected user
      let uid = $(".selectpicker").selectpicker('val')

      // If the value was changed to blank, ignore it
      if (uid == "") return

      let currentListItems = $("#user-list li").toArray().map(user => $(user).data("uid"))

      // Check to make sure that the user is not already there
      if (!currentListItems.includes(uid)) {
```

```
    // Create the faded list item and display it
    let option = $("option[value='" + uid + "']")

    let listItem = document.createElement("li")
    listItem.setAttribute('class', 'list-group-item text-left not-saved add-access');
    listItem.setAttribute("data-uid", uid)

    let spanItemAvatar = document.createElement("span"),
        spanItemName = document.createElement("span"),
        spanItemUser = document.createElement("span");
    spanItemAvatar.setAttribute('class', 'avatar float-left mr-2');
    spanItemAvatar.innerText = option.text().charAt(0);
    spanItemName.setAttribute('class', 'shared-user');
    spanItemName.innerText = option.text();
    spanItemUser.setAttribute('class', 'text-muted');
    spanItemUser.innerText = option.data('subtext');
    spanItemName.append(spanItemUser);

    listItem.innerHTML = "<span class='text-primary float-right shared-user cursor-pointer' on
    listItem.prepend(spanItemName);
    listItem.prepend(spanItemAvatar);

    $("#user-list").append(listItem)
  }
})

$("#presentation-upload").change(function(data) {
  var file = data.target.files[0]

  // Check file type and size to make sure they aren't over the limit
  if (validFileUpload(file)) {
    $("#presentation-upload-label").text(file.name)
  } else {
    $("#invalid-file-type").show()
    $("#presentation-upload").val("")
    $("#presentation-upload-label").text($("#presentation-upload-label").data("placeholder"))
  }
})

$(".preupload-room").click(function() {
  updatePreuploadPresentationModal(this)
})

$("#remove-presentation").click(function(data) {
  removePreuploadPresentation($(this).data("remove"))
})

// trigger initial room filter
filterRooms();
```

```
  }
});

function copyInvite() {
  $('#invite-url').select()
  if (document.execCommand("copy")) {
    $('#invite-url').blur();
    copy = $("#copy-invite")
    copy.addClass('btn-success');
    copy.html("<i class='fas fa-check mr-1'></i>" + getLocalizedString("copied"))
    setTimeout(function(){
      copy.removeClass('btn-success');
      copy.html("<i class='fas fa-copy mr-1'></i>" + getLocalizedString("copy"))
    }, 1000)
  }
}

function copyAccess(target) {
  input = target ? $("#copy-" + target + "-code") : $("#copy-code")
  input.attr("type", "text")
  input.select()
  if (document.execCommand("copy")) {
    input.attr("type", "hidden")
    copy = target ? $("#copy-" + target + "-access") : $("#copy-access")
    copy.addClass('btn-success');
    copy.html("<i class='fas fa-check mr-1'></i>" + getLocalizedString("copied"))
    setTimeout(function(){
      copy.removeClass('btn-success');
      originalString = target ? getLocalizedString("room.copy_" + target + "_access") : getLocaliz
      copy.html("<i class='fas fa-copy mr-1'></i>" + originalString)
    }, 1000)
  }
}

function showCreateRoom(target) {
  $("#create-room-name").val("")
  $("#create-room-access-code").text(getLocalizedString("modal.create_room.access_code_placeholder
  $("#create-room-moderator-access-code").text(getLocalizedString("modal.create_room.moderator_acc
  $("#room_access_code").val(null)
  $("#room_moderator_access_code").val(null)

  $("#createRoomModal form").attr("action", $("body").data('relative-root'))
  $("#room_mute_on_join").prop("checked", $("#room_mute_on_join").data("default"))
  $("#room_require_moderator_approval").prop("checked", $("#room_require_moderator_approval").data
  $("#room_anyone_can_start").prop("checked", $("#room_anyone_can_start").data("default"))
  $("#room_all_join_moderator").prop("checked", $("#room_all_join_moderator").data("default"))
  $("#room_recording").prop("checked", $("#room_recording").data("default"))

  //show all elements & their children with a create-only class
  $(".create-only").each(function() {
    $(this).show()
    if($(this).children().length > 0) { $(this).children().show() }
  })

  //hide all elements & their children with a update-only class
  $(".update-only").each(function() {
    $(this).attr('style',"display:none !important")
    if($(this).children().length > 0) { $(this).children().attr('style',"display:none !important")
  })
}

function showUpdateRoom(target) {
  var modal = $(target)
  var update_path = modal.closest(".room-block").data("path")
  var settings_path = modal.data("settings-path")
  $("#create-room-name").val(modal.closest(".room-block").find(".room-name-text").text().trim())
  $("#createRoomModal form").attr("action", update_path)

  //show all elements & their children with a update-only class
  $(".update-only").each(function() {
    $(this).show()
    if($(this).children().length > 0) { $(this).children().show() }
  })

  //hide all elements & their children with a create-only class
  $(".create-only").each(function() {
    $(this).attr('style',"display:none !important")
    if($(this).children().length > 0) { $(this).children().attr('style',"display:none !important")
  })

  updateCurrentSettings(settings_path)

  var accessCode = modal.closest(".room-block").data("room-access-code")

  if(accessCode){
    $("#create-room-access-code").text(getLocalizedString("modal.create_room.access_code") + ": "
    $("#room_access_code").val(accessCode)
  } else {
    $("#create-room-access-code").text(getLocalizedString("modal.create_room.access_code_placehold
    $("#room_access_code").val(null)
  }

  var moderatorAccessCode = modal.closest(".room-block").data("room-moderator-access-code")

  if(moderatorAccessCode){
    $("#create-room-moderator-access-code").text(getLocalizedString("modal.create_room.moderator_a
    $("#room_moderator_access_code").val(moderatorAccessCode)
  } else {
    $("#create-room-moderator-access-code").text(getLocalizedString("modal.create_room.moderator_a
    $("#room_moderator_access_code").val(null)
  }
}

function showDeleteRoom(target) {
  $("#delete-header").text(getLocalizedString("modal.delete_room.confirm").replace("%{room}", $(ta
  $("#delete-confirm").parent().attr("action", $(target).data("path"))
}

//Update the createRoomModal to show the correct current settings
function updateCurrentSettings(settings_path){
  // Get current room settings and set checkbox
  $.get(settings_path, function(settings) {
    $("#room_mute_on_join").prop("checked", $("#room_mute_on_join").data("default") || settings.mu
    $("#room_require_moderator_approval").prop("checked", $("#room_require_moderator_approval").da
    $("#room_anyone_can_start").prop("checked", $("#room_anyone_can_start").data("default") || set
    $("#room_all_join_moderator").prop("checked", $("#room_all_join_moderator").data("default") ||
    $("#room_recording").prop("checked", $("#room_recording").data("default") || Boolean(settings.
  })
}

function generateAccessCode(){
  const accessCodeLength = 6
  var validCharacters = "0123456789"
  var accessCode = ""

  for( var i = 0; i < accessCodeLength; i++){
    accessCode += validCharacters.charAt(Math.floor(Math.random() * validCharacters.length));
  }

  $("#create-room-access-code").text(getLocalizedString("modal.create_room.access_code") + ": " +
  $("#room_access_code").val(accessCode)
}

function ResetAccessCode(){
  $("#create-room-access-code").text(getLocalizedString("modal.create_room.access_code_placeholder
  $("#room_access_code").val(null)
}

function generateModeratorAccessCode(){
  const accessCodeLength = 6
  var validCharacters = "abcdefghijklmopqrstuvwxyz"
  var accessCode = ""

  for( var i = 0; i < accessCodeLength; i++){
    accessCode += validCharacters.charAt(Math.floor(Math.random() * validCharacters.length));
  }

```



```
  $("#create-room-moderator-access-code").text(getLocalizedString("modal.create_room.moderator_acc
  $("#room_moderator_access_code").val(accessCode)
}

function ResetModeratorAccessCode(){
  $("#create-room-moderator-access-code").text(getLocalizedString("modal.create_room.moderator_acc
  $("#room_moderator_access_code").val(null)
}

function saveAccessChanges() {
  let listItemsToAdd = $("#user-list li:not(.remove-shared)").toArray().map(user => $(user).data("

  $.post($("#save-access").data("path"), {add: listItemsToAdd})
}

// Get list of users shared with and display them
function displaySharedUsers(path) {
  $.get(path, function(users) {

    $("#user-list").html("") // Clear current inputs

    users.forEach(function(user) {

      listName = document.createElement("li"),
      spanAvatar = document.createElement("span"),
      spanName = document.createElement("span"),
      spanUid = document.createElement("span"),
      spanRemove = document.createElement("span"),
      spanRemoveIcon = document.createElement("i");

      listName.setAttribute('class', 'list-group-item text-left')
      listName.setAttribute('data-uid', user.uid)
      spanAvatar.innerText = user.name.charAt(0)
      spanAvatar.setAttribute('class', 'avatar float-left mr-2')
      spanName.setAttribute('class', 'shared-user')
      spanName.innerText = user.name
      spanUid.setAttribute('class', 'text-muted ml-1')
      spanUid.innerText = user.uid
      spanRemove.setAttribute('class', 'text-primary float-right shared-user cursor-pointer')
      spanRemove.setAttribute('onclick', 'removeSharedUser(this)')
      spanRemoveIcon.setAttribute('class', 'fas fa-times')

      listName.appendChild(spanAvatar)
      listName.appendChild(spanName)
      spanName.appendChild(spanUid)
      listName.appendChild(spanRemove)
      spanRemove.appendChild(spanRemoveIcon)

      $("#user-list").append(listName)
    })
  });
}

// Removes the user from the list of shared users
function removeSharedUser(target) {
  let parentLI = target.closest("li")

  if (parentLI.classList.contains("not-saved")) {
    parentLI.parentNode.removeChild(parentLI)
  } else {
    parentLI.removeChild(target)
    parentLI.classList.add("remove-shared")
  }
}

function updatePreuploadPresentationModal(target) {
  $.get($(target).data("settings-path"), function(presentation) {
    if(presentation.attached) {
      $("#current-presentation").show()
      $("#presentation-name").text(presentation.name)
      $("#change-pres").show()
      $("#use-pres").hide()
    } else {
      $("#current-presentation").hide()
      $("#change-pres").hide()
      $("#use-pres").show()
    }
  });

  $("#preuploadPresentationModal form").attr("action", $(target).data("path"))
  $("#remove-presentation").data("remove",  $(target).data("remove"))

  // Reset values to original to prevent confusion
  $("#presentation-upload").val("")
  $("#presentation-upload-label").text($("#presentation-upload-label").data("placeholder"))
  $("#invalid-file-type").hide()
}

function removePreuploadPresentation(path) {
  $.post(path, {})
}

function validFileUpload(file) {
  return file.size/1024/1024 <= 30
}

// Automatically click the join button if this is an action cable reload
function checkIfAutoJoin() {
  var url = new URL(window.location.href)

  if (url.searchParams.get("reload") == "true") {
    $("#joiner-consent").click()
    $("#room-join").click()
  }
}

function filterRooms() {
  let search = $('#room-search').val()

  if (search == undefined) { return }

  let search_term = search.toLowerCase(),
        rooms = $('#room_block_container > div:not(:last-child)');
        clear_room_search = $('#clear-room-search');

  if (search_term) {
    clear_room_search.show();
  } else {
    clear_room_search.hide();
  }

  rooms.each(function(i, room) {
    let text = $(this).find('h4').text();
    room.style.display = (text.toLowerCase().indexOf(search_term) < 0) ? 'none' : 'block';
  })
}

function clearRoomSearch() {
  $('#room-search').val('');
  filterRooms()
}

function manageAccessAccessibility() {
  // share room pop up accessibility
  var holdModal = false;
  $("#shareRoomModal").on("show.bs.modal", function() {
    // for screen reader to be able to read results
    $("#shareRoomModal .form-control").attr("aria-atomic", true);
    $("#shareRoomModal .dropdown-menu div.inner").attr("role", "alert");
    $("#shareRoomModal ul.dropdown-menu").attr("role", "listbox");
    $("#shareRoomModal div.dropdown-menu").find("*").keyup(function(event) {
      $("#shareRoomModal ul.dropdown-menu li").attr("aria-selected", false);
      $("#shareRoomModal ul.dropdown-menu li.active").attr("aria-selected", true);
      $("#shareRoomModal ul.dropdown-menu li.active a").attr("aria-selected", true);
    });
    // for keyboard support
    // so that it can escape / close search user without closing the modal
    $("#shareRoomModal div.dropdown-menu input").keydown(function(event) {
      if (event.keyCode === 27) {
        holdModal = true;
      }
    });
  });

  // reset escape button if the search is closed / done
  $("#shareRoomModal").on("hide.bs.modal", function(e) {
    if (holdModal) {
      holdModal = false;
      e.stopPropagation();
      return false;
    }
  });
}

function generateAccessCodeAccessibility() {
  // For keyboard users to be able to generate access code
  $("#generate-room-access-code").keyup(function(event) {
    if (event.keyCode === 13 || event.keyCode === 32) {
        generateAccessCode();
    }
  })

  // For keyboard users to be able to reset access code
  $("#reset-access-code").keyup(function(event) {
    if (event.keyCode === 13 || event.keyCode === 32) {
        ResetAccessCode();
    }
  })

  // For keyboard users to be able to generate access code
  // for moderator
  $("#generate-moderator-room-access-code").keyup(function(event) {
    if (event.keyCode === 13 || event.keyCode === 32) {
        generateModeratorAccessCode();
    }
  })

  // For keyboard users to be able to reset access code
  // for moderator
  $("#reset-moderator-access-code").keyup(function(event) {
    if (event.keyCode === 13 || event.keyCode === 32) {
        ResetModeratorAccessCode();
    }
  })
}
```