# Cấu trúc data gửi request đến Server

url: “php-controller/ServerHandler.php”

type: {GET/POST}

data: {

role : { guest/ user }

operation: {getPartner/getXXX}

sentData:{

…  
 }

}

# Validate input

Viết trong file common.js như

* validateEmail : format email
* validateText: chỉ cho phép các kí tự chữ cái và số
* validateModal: username , password , email v.v..

# Hỗ trợ multi-language ( dùng JS )

Trong file lang.js

**var** msg = {  
 en:{  
 fill\_in\_user\_name\_field : "Please fill in 'User Name' field",  
 fill\_in\_pass\_field: "Please fill in 'Password' field",  
 fill\_in\_email\_field: "Please fill in 'Email' field",  
 fill\_in\_name\_field: "Please fill in 'Name' field",  
 invalid\_email\_format: "Invalid email format",  
 name\_field\_allow\_letters\_numbers : "'Name' field only allowed letters and numbers",  
 user\_name\_field\_allow\_letters\_numbers: "'User Name' field only allowed letters and numbers",  
 pass\_field\_allow\_letters\_numbers: "'Password' field only allowed letters and numbers",  
 birth\_place\_field\_allow\_letters\_numbers: "'Password' field only allowed letters and numbers",  
 confirm\_pass\_not\_match : "The confirmation password does not match the password you first entered",  
 user\_or\_pass\_incorrect: "User name or password is incorrect",  
 user\_already\_exist: "User already existed. Please try new username or email instead!",  
 fill\_in\_birth\_place\_field:"Please fill in 'BirthPlace' field",  
 child\_can\_not\_old\_than\_parent:"Child can't be older than parent",  
 user\_name\_field\_is\_too\_long: "'User name' must be less than 25 characters",  
 pass\_field\_is\_too\_long: "'Password' must be less than 25 characters",  
 name\_field\_is\_too\_long: "'Name' must be less than 25 characters",  
 email\_field\_is\_too\_long: "'Email' must be less than 25 characters",  
 birthplace\_field\_is\_too\_long: "'Birthplace' must be less than 25 characters"  
 },  
 vi:{  
 fill\_in\_user\_name\_field : "Hãy nhập vào 'Tên người dùng'",  
 fill\_in\_pass\_field: "Hãy nhập vào 'Mật khẩu'",  
 fill\_in\_email\_field: "Hãy nhập vào Email'",  
 fill\_in\_name\_field: "Hãy nhập vào 'Tên'",  
 invalid\_email\_format: "Định dạng email không đúng",  
 name\_field\_allow\_letters\_numbers: "'Name' chỉ được chứa kí tự và số",  
 user\_name\_field\_allow\_letters\_numbers: "'Tên người dùng' chỉ được chứa kí tự và số",  
 pass\_field\_allow\_letters\_numbers: "'Mật khẩu' chỉ được chứa kí tự và số",  
 confirm\_pass\_not\_match : "Mật khẩu nhập lại không trùng khớp",  
 user\_or\_pass\_incorrect: "Tài khoản và mật khẩu không trùng khớp",  
 user\_already\_exist: "Tài khoản đã tồn tại. Bạn hãy sử dụng tài khoản hay email khác!",  
 fill\_in\_birth\_place\_field: "Hãy nhập vào 'Nơi sinh'",  
 child\_can\_not\_old\_than\_parent: "Con không thể lớn hơn bố/mẹ",  
 birth\_place\_field\_allow\_letters\_numbers: "'Nơi sinh' chỉ được chứa kí tự và số",  
 user\_name\_field\_is\_too\_long: "'Tên người dùng' dưới 25 kí tự",  
 pass\_field\_is\_too\_long:"'Mật khẩu' dưới 25 kí tự",  
 email\_field\_is\_too\_long: "'Email' dưới 25 kí tự",  
 birthplace\_field\_is\_too\_long: "'Nơi sinh' dưới 25 kí tự",  
 name\_field\_is\_too\_long: "'Tên' dưới 25 kí tự"  
 }  
};

Sử dụng:

msg[lang][“user\_name\_field\_is\_too\_long”]

# Up ảnh avatar thông qua host free imgur

Sử dụng ajax kết nối với host imgur để đưa file ảnh từ client lên host. Bằng cách này tiết kiệm được không gian lưu trữ trên máy chủ.

$("#btnUploadAvatar").click(**function**(){  
 memberUploadAvatarId = $(**this**).attr("data-memid");  
 avatarId = $(**this**).attr("data-avatarid");  
 isPartner = $(**this**).attr("data-isPartner");  
 console.log(isPartner);  
 **var** isAddMem = $(**this**).attr("data-addmem");  
 **if** (imgUrl != **null** && imgUrl != "") {  
 $.ajax({  
 url: "https://api.imgur.com/3/upload",  
 type: "POST",  
 dataType: "json",  
 data: {image: imgUrl},  
 success: **function**(data) {  
 updateAvatarForDB(data, isAddMem);  
  
 },  
 error: showMeError,  
 beforeSend: **function** (xhr) {  
 $('#modal-uploading').modal('show');  
 xhr.setRequestHeader("Authorization", "Client-ID " + clientId);  
 }  
 });  
 }  
});

# Quản lý guest

File management.php , guests-management.js

3 modal quản lý : thêm , xóa , sửa guest

Thoạt đầu thì file .js sẽ load toàn bộ guest:

*/\*\*  
 \* Load all users  
 \*/* $.ajax({  
 url: 'php-controller/ServerHandler.php',  
 type: 'GET',  
 dataType: 'json',  
 data:{  
 role : "user",  
 operation: "getAllGuests",  
 UserID: $("head").prop('data-id')  
 }  
 }).done(**function**(data){  
$('#modal-uploading').modal('hide');  
 createTable(data);  
 search(data);  
 console.log("Load guests successfully");  
 }).fail(**function** () {  
$('#modal-uploading').modal('hide');  
 console.log("Load guests failed");  
 });

# Xác thực người dùng

Khi người dùng đăng nhập và được xác nhận là đúng username – password, server sẽ trả về một object, gọi là token. Token là một chuỗi gồm id (trong database), username và role đã được mã hóa và lưu trong cookie. Để mã hóa chuỗi id, username và role thành token, ở đây ta sử dụng thư viện php-jwt, được tải về thông qua Composer. Thư viện này cung cấp các hàm mã hóa và giải mã token thông qua các giải thuật thông dụng. Ở đây ta sử dụng HS512.

Mỗi khi người dùng gởi một request lên server, token sẽ được tự động gửi kèm cùng request. Ta sẽ sử dụng token này để xác thực danh tính người dùng đã gởi. Đây là code kiểm tra token:

if (isset($\_COOKIE['token'])) {

$token = $\_COOKIE['token'];

$data = (array) JWT::decode($token, Token::$jwt\_key,

['alg' => 'HS512']);

if (!$data) {

header('HTTP/1.1 400 Bad Request');

exit;

}

…

}

Khi đã xác thực đúng, ta sẽ sử dụng dữ liệu được giải mã từ token để tiến hành truy vấn dữ liệu tùy theo request của người dùng.

Ngoài ra, token còn được sử dụng để xem xét quyền và điều hướng người dùng khi truy cập một trang web nhất định. Chẳng hạn khi một người dùng (đã đăng nhập) nhập vào thanh địa chỉ đường dẫn dẫn đến trang chủ (/ hoặc /index) thì server sẽ tự động điều hướng người dùng vào trang /tree (đối với role “user” hoặc “guest”) hay /management (đối với role “admin”). Đoạn code như sau:

if (isset($\_COOKIE['token'])) {

$token = $\_COOKIE['token'];

$data = (array) JWT::decode($token, Token::$jwt\_key, ['alg'

=> 'HS512']);

$personData = (array) $data['data'];

$id = $personData['id'];

$role = $personData['role'];

if ($role == 'admin') header('Location: management');

else header('Location: tree');

}

Nếu người dùng với role “admin” truy cập vào đường dẫn /tree thì sẽ nhận được dòng chữ “You are not allowed to be here” do admin chỉ có nhiệm vụ quản lý các user. Tương tự guest không thể truy cập đến đường dẫn /management do họ chỉ được phép xem gia phả do một user nào đó tạo ra và có quyền quản lý đối với guest đó.

# Kéo thả cây gia phả

Để rà soát cây gia phả gia đình mình, người dùng có thể kéo gia phả lên xuống hoặc trái phải. Để thực hiện điều này, ta sử dụng thư viện interactjs. Đoạn code thực hiện điều này được viết ở trong file tree.js như sau:

interact('.tree').draggable({

// enable inertial throwing

inertia: true,

// keep the element within the area of its parent

restrict: {

restriction: "parent",

endOnly: true,

elementRect: { top: 0 + heightOffset, left: 0 + widthOffset, bottom: 1 - heightOffset, right: 1- widthOffset }

},

// enable autoScroll

autoScroll: true,

// call this function on every dragmove event

onmove: function (event) {

var target = event.target,

// keep the dragged position in the data-x/data-y attributes

x = (parseFloat(target.getAttribute('data-x')) || 0) + event.dx,

y = (parseFloat(target.getAttribute('data-y')) || 0) + event.dy;

// translate the element

target.style.webkitTransform =

target.style.transform =

'translate(' + x + 'px, ' + y + 'px)';

// update the posiion attributes

target.setAttribute('data-x', x);

target.setAttribute('data-y', y);

}

});

Các biến heightOffset và widthOffset được tính toán để đảm bảo rằng cây gia phả luôn nằm trong giới hạn trang web. Hàm onmove để lưu lại vị trí của cây gia phả mà ta đã kéo đến.

# Đa ngôn ngữ

Trang web hỗ trợ UI tiếng Anh và tiếng Việt. Để hiện đúng các dòng chữ trên UI tùy theo ngôn ngữ, ta sẽ sử dụng các hàm PHP. Ở đây, ta viết class I18N với thuộc tính là ngôn ngữ được người dùng chọn (“en” hoặc “vi”). Trong class ta sẽ viết các hàm dựa trên thuộc tính nói trên để trả về đúng dòng chữ cần hiện.

Mỗi khi trang web được load, ta sẽ kiểm tra cookie để biết ngôn ngữ mà người dùng đang sử dụng là gì và trả về UI trong ngôn ngữ đó. Trong trường hợp người dùng lần đầu sử dụng, chưa từng thiết đặt ngôn ngữ thì ta mặc định là tiếng Anh. Đoạn code như sau:

if (!isset($\_COOKIE['lang'])) {

setcookie('lang', 'en');

$i18n = new i18n("en");

}

else {

$i18n = new i18n($\_COOKIE['lang']);

}

Người dùng sẽ chọn ngôn ngữ mình muốn sử dụng thông qua hai icon lá cờ ở góc trên bên phải. Ngôn ngữ được chọn sẽ được lưu vào cookie, sau đó trang web sẽ được reload để cập nhật lại UI sang ngôn ngữ mới.

$('.language li:not(.active)').click(function() {

setCookie('lang', $(this).attr('data-lang'), 30);

document.location.reload();

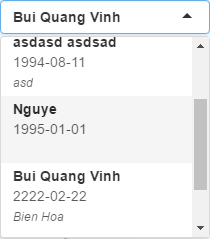
});

# Tìm kiếm thành viên trong cây gia phả

Cho phép người dùng khả năng tìm kiếm một thành viên đang có mặt trong cây gia phả dựa vào các thông tin của thành viên:

* Tên thành viên
* Email thành viên
* Địa chỉ của thành viên
* Ngày sinh của thành viên
* Giới tình của thành viên





Các thành viên có những thông tin phù hợp với thông tin người dùng cần tìm kiếm sẽ được làm nổi bật lên.như hình sau



Nhóm sử dụng thư viện selectizejs để hỗ trợ hiện thức chức năng tìm kiếm. Và đây là các bước hiện thực chức năng

* Bước đầu tiên là xác định những thông tin mà người dùng có thể sử dụng để tìm kiếm

searchField: ['Name', "Gender", 'Address', 'BirthDate', 'BirthPlace']

* Bước thứ hai là render các option và item của công cụ tìm kiếm.

render: {

item: function(item, escape) {

return '<div><strong>' + escape(item.Name) + '</strong></div>';

},

option: function(item, escape) {

return '<div data-id="' + escape(item.MemberID) + '"><strong>' +

escape(item.Name) +

'</strong><br><span style="opacity:0.8;">' + escape(item.BirthDate) +

'</span><br><small style="font-style: italic; opacity:0.8;">' +

escape(item.Address) + '</small><br></div>';

}

}

* Bước thứ ba, làm các member trong cây giả phả nổi bậc lên sau khi người dùng thay đổi thông tin tìm kiếm.

onType: function(str){

text = str;

$('.membercard').removeClass('border-effect');

if(str){

$('.selectize-dropdown .selectize-dropdown-content div').each( function(){

$('#mem' + $(this).attr('data-value')).addClass('border-effect');

});

}

},