

A guide on how to add new module on the server. The following are the steps to follow when adding new server module.

1. Register the module in modules list. This is only required if the module communicates with the agent. Modules that are present only on the server side do not need this.

File to edit: `/includes/const.js`

Edit here:

```
exports.messageKeys = {
  call: '0xCL',
  clipboard: '0xCB',

  ...

  [NEW_MODULE_NAME]: [NEW_MODULE_ID]
}
```

2. Register the module in client database. This is required to store the output received from agents to client database and later retrieve it for display. Server-only modules also require to register here if they have data that needs to be displayed or processed. For instance, command queue module stores commands queued to the client in database to send them later when the agent becomes online, and it's also displayable to the user.

File to edit: `/includes/databaseGateway.js`

Edit here:

```
class clientdb {
  constructor(clientID) {
    let cdb = lowdb(new FileSync('./clientData/' + clientID + '.json'))
    cdb.defaults({
      clientID,
      CallData: [],
      clipboardLog: [],
      CommandQueue: [],

      ...

      [NEW_MODULE_DB_KEY]: [JSON HOLDER (Array, Object, ...)]
    })
  }
}
```

3. Create the module page. Create the page that will display the data of the module.

Add file here:

[/assets/views/deviceManagerPages/NEW_PAGE_NAME.ejs](#)

4. Register the created module page. The page created in step (3) needs to be registered so that it's possible to redirect to it when it's menu in step (5) is clicked.

File to edit: [/assets/views/deviceManager.ejs](#)

Edit here:

```
<div class="ui container">
  <% include partials/header.ejs %>
  <div class="ui grid">
    <% include partials/deviceManagerMenu.ejs %>
    <div class="twelve wide column">
      <script>
        gtag('event', 'screen_view', {
          'screen_name': '<%= page %>',
          'app_name': 'L3MON'
        });
      </script>
      <%
        if(page === "calls") { %><% include deviceManagerPages/call_log.ejs %><% }
        else if(page === "clipboard") { %><% include deviceManagerPages/clipboard_log.ejs %><% }
        else if(page === "commands") { %><% include deviceManagerPages/commands.ejs %><% }

        ...

        else if(page === "[NEW_PAGE_NAME]") { %><% include
          deviceManagerPages/[NEW_PAGE_NAME.ejs] %><% }
      %>
    </div>
  </div>
</div>
```

5. Register modules menu on the main agent UI. This menu will be visible on the main UI of an agent page.

File to edit: [/assets/views/partial/deviceManagerMenu.ejs](#)

Edit here:

```
<div class="four wide column">
```

```

<div class="ui vertical pointing secondary fluid menu hoverEdit">
  <a href="<%= baseURL %>/calls" class="item <% if (page=== 'calls') { %> active <% } %>">
    <i class="icon phone grey"></i> Call Log
  </a>
  <a href="<%= baseURL %>/clipboard" class="item <% if (page=== 'clipboard') { %> active <% } %>">
    <i class="icon clipboard grey"></i> Clipboard Log
  </a>
  <a href="<%= baseURL %>/commands" class="item <% if (page=== 'commands') { %> active <% } %>">
    <i class="icon download grey"></i> Command Queue
  </a>

  ...

  <a href="<%= baseURL %>/[NEW_PAGE_NAME]" class="item <% if (page=== '[NEW_PAGE_NAME]') { %> active <% } %>">
    <i class="icon [ICON_NAME] alternate grey"></i> NEW_PAGE_DISPLAY_TEXT
  </a>
</div>
</div>

```

6. Add socket handler to receive and store evidence from an agent to database. This is only required if the module communicates with the agent. Modules that are present only on the server side do not need this.

File to edit: `/includes/clientManager.js`

Edit here:

```

setupListeners(clientID) {
  let socket = this.clientConnections[clientID];
  let client = this.getClientDatabase(clientID);

  ...

  socket.on(CONST.messageKeys.[NEW_MODULE_NAME], (data) => {
    if (data.[DATA_ROOT_JSON_KEY]) {
      if (data.[DATA_ROOT_JSON_KEY].length !== 0) {
        let list = data.[DATA_ROOT_JSON_KEY];
        let dbCall = client.get('EventData');
        let newCount = 0;
        list.forEach(item => {
          let hash =
            crypto.createHash('md5').update(item.[JSON_DATA_FIELD_HASH_KEY]).digest("hex");
          if (dbCall.find({ hash }).value() === undefined) {
            // cool, we dont have this item
            item.hash = hash;

```

```

        dbCall.push(item).write();
        newCount++;
    }
    });
    LogManager.log(CONST.logTypes.success, clientID + " [NEW_MODULE_NAME] Log Updated - "
        + newCount + " [NEW_MODULE_NAME]");
}
}
});
}

```

7. Add menu handler to populate the page with data, when a menu is clicked. This will be responsible to populate data from database to the page.

File to edit: `/includes/clientManager.js`

Edit here:

```

getClientDataByPage(clientID, page, filter = undefined) {
    let client = db.maindb.get('clients').find({ clientID }).value();
    if (client !== undefined) {
        let clientDB = this.getClientDatabase(client.clientID);
        let clientData = clientDB.value();

        let pageData;

        if (page === "calls") {
            pageData = clientDB.get('CallData').sortBy('date').reverse().value();
            if (filter) {
                let filterData = clientDB.get('CallData').sortBy('date').reverse().value().filter(calls =>
                    calls.phoneNo.substr(-6) === filter.substr(-6));
                if (filterData) pageData = filterData;
            }
        }
        else if (page === "clipboard") pageData =
            clientDB.get('clipboardLog').sortBy('time').reverse().value();
        else if (page === "commands") pageData = clientData.CommandQue;

        ...

        else if (page === "[NEW_PAGE_NAME]") pageData =
            clientDB.get('[NEW_MODULE_DB_KEY]').sortBy('[JSON_DATA_FIELD_SORT_KEY]').reverse().value();

        return pageData;
    } else return false;
}

```

8. Add error checking for command configuration params. If the server module sends additional command configuration along with the module command, add missing param error checking for the parameters. For example, a microphone record module sends the duration of the record in seconds.

File to edit: `/includes/clientManager.js`

Edit here:

```
checkCorrectParams(commandID, commandPayload, cb) {  
  if (commandID === CONST.messageKeys.mic) {  
    if (!'sc' in commandPayload)  
      return cb('Mic Missing `sec` Parameter')  
    else cb(false)  
  }  
  
  ...  
  
  else if (commandID === [NEW_MODULE_NAME]) {  
    if (!'CMD_PARAM_JSON_KEY' in commandPayload)  
      return cb('[PARAMETER MISSING ERROR MESSAGE TO USER')  
    else cb(false)  
  }  
}
```