**TÀI LIỆU KỸ THUẬT HỆ THỐNG**

**WEBSITE BÁO CÁO TÀI CHÍNH 2021**

|  |  |  |  |
| --- | --- | --- | --- |
| **PHÊ DUYỆT** | | | |
| **Mục** | **Biên soạn** | **Kiểm tra** | **Phê duyệt** |
| Ngày/tháng/năm | ……../……../20…….. | ……../……../20…….. | ……../……../20…….. |
| Chức danh | **CV NOC** | **TP. VHDV** | **GĐ. KCN** |
| Chữ ký |  |  |  |
| Họ và tên |  |  |  |

**BẢNG KIỂM SOÁT QUÁ TRÌNH THAY ĐỔI**

|  |  |  |  |
| --- | --- | --- | --- |
| **Soát xét** | **Ngày hiệu lực** | **Người chỉnh sửa** | **Nội dung thay đổi** |
| 00 | 05/11/2021 | Trần Minh Quân | Biên soạn lần đầu |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**MỤC LỤC**

[I. GIỚI THIỆU 4](#_Toc87019865)

[1. Giới thiệu chung 4](#_Toc87019866)

[2. Mục đích và đối tượng sử dụng 4](#_Toc87019867)

[3. Một số định nghĩa và từ viết tắt 4](#_Toc87019868)

[4. Tài liệu tham khảo 4](#_Toc87019869)

[II. KIẾN TRÚC HỆ THỐNG 4](#_Toc87019870)

[1. Cấu trúc hệ thống báo cáo tài chính 2021 4](#_Toc87019871)

[2. Giải pháp công nghệ 5](#_Toc87019872)

[3. Các thành phần của hệ thống 5](#_Toc87019873)

[4. Mô hình kết nối 6](#_Toc87019874)

[5. Luồng xử lý: 6](#_Toc87019875)

[III. THAM SỐ VÀ LOG FILE 6](#_Toc87019876)

[1. Tham số hệ thống 6](#_Toc87019877)

[2. Các file Log hệ thống 7](#_Toc87019878)

[IV. CÀI ĐẶT HỆ THỐNG 7](#_Toc87019879)

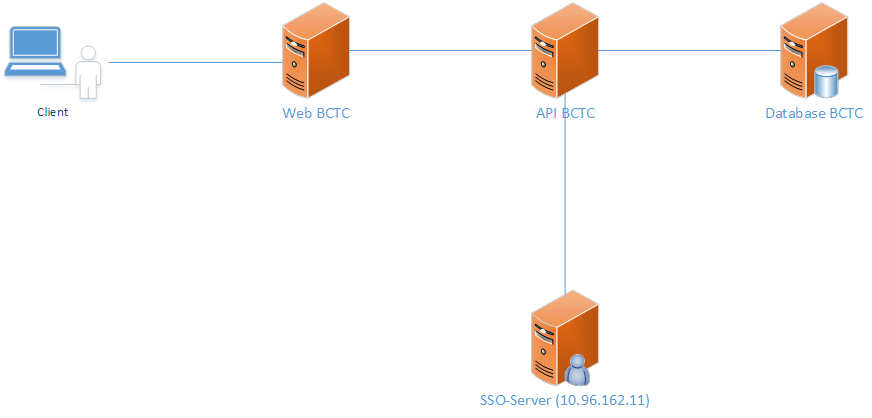
[1. Cài đặt API 7](#_Toc87019880)

[2. Cài đặt Web BCTC 12](#_Toc87019881)

1. GIỚI THIỆU
2. Giới thiệu chung

Hệ thống CIC H2H dung để truy vấn và lưu trữ thông tin khách hang từ trung tâm CIC cho nội bộ OCB sử dụng

1. Mục đích và đối tượng sử dụng
2. Một số định nghĩa và từ viết tắt
3. Tài liệu tham khảo
4. KIẾN TRÚC HỆ THỐNG
5. Cấu trúc hệ thống báo cáo tài chính 2021



1. Giải pháp công nghệ

Website CIC và API CIC sử dụng mô hình công nghệ .Net Core MVC 3.1, Entity framework,

Hệ thống xác thực user thông qua LDAP sử dụng NodeJS (SSO Server).

Source Core gốc được phát triển bởi quantm1@ocb.com.vn

1. Các thành phần của hệ thống

**API BCTC:**

* Xác thực user đăng nhập thông qua SSO SERVER
* Giao tiếp với API Microfund để lấy thông tin.
* Nhận file upload và lưu trữ vào server dưới dạng file upload.
* Lưu trữ và xử lý dữ liệu lấy được từ Microfund vào DB SQL

**WEB BCTC:**

* Giao diện nghiệp vụ cho user
* Giao tiếp với API BCTC để xác thực người dung, lấy dữ liệu.

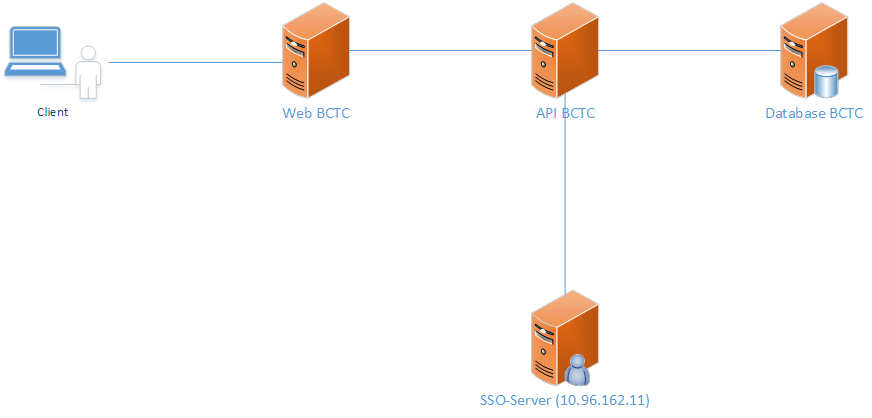
**SSO SERVER:**

* Xác thực người dùng thông qua LDAP
* Gửi socket notification

**DB SQL BÁO CÁO TÀI CHÍNH WEB:**

* Lưu trữ dữ liệu lấy được từ Microfund
* Lưu trữ dữ liệu master của hệ thống
* Lưu trữ các xử lý nghiệp vụ

1. Mô hình kết nối



1. Luồng xử lý:

* Người dùng tiến hành login vào hệ thống để sử dụng. API nhận được thông tin và tiến hành xác thực người dùng trên LDAP thông qua SSO Server.
* Người dùng thao tác nghiệp vụ trên WEB BCTC và tiến hành submit request
* API BCTC nhận được thông tin xử lý, tiến hành xử lý dữ liệu và connect đến Microfund để lấy thông tin
* Sau khi nhận được kết quả trả về từ Microfund, API tiến hành xử lý dữ liệu và lưu trữ xuống database, đồng thời trả kết quả cuối cùng về cho WEB BCTC

1. THAM SỐ VÀ LOG FILE
2. Tham số hệ thống

API:

* Key xác thực certificate của website BCTC
* Key xác thực của các hệ thống khác
* Key xác thực để gọi qua SSO
* Key xác thực để gọi qua Microfund

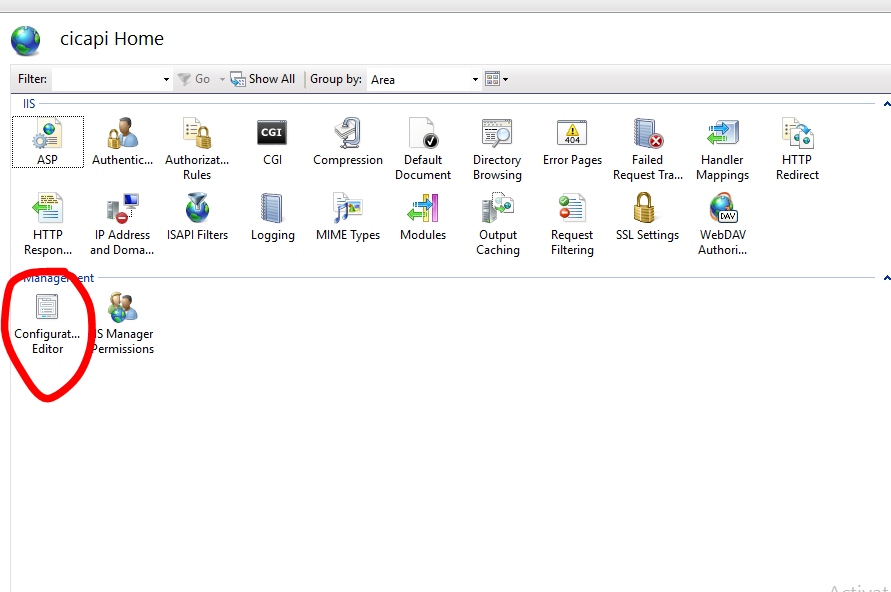
1. Các file Log hệ thống

API:

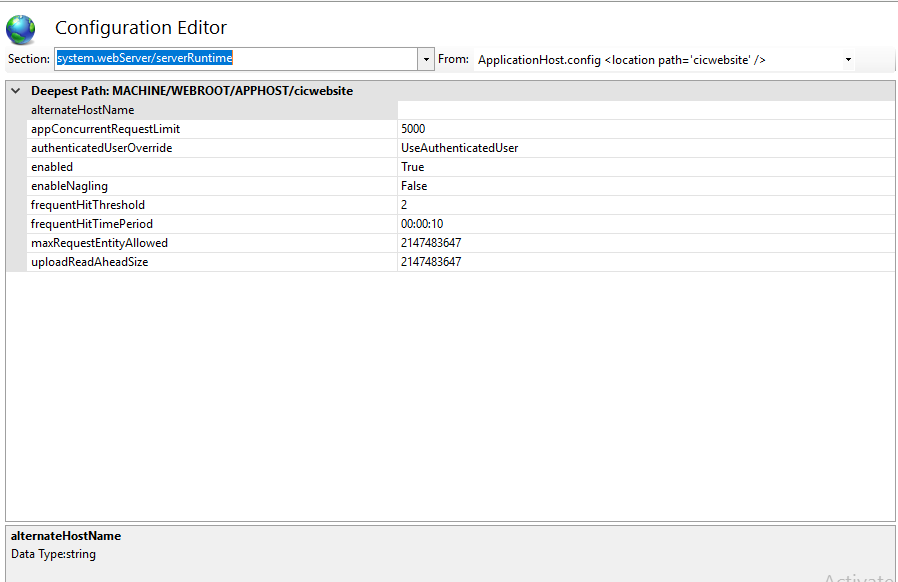
* Log input/error: nằm ở thư mục Log

1. CÀI ĐẶT HỆ THỐNG
2. Cài đặt API

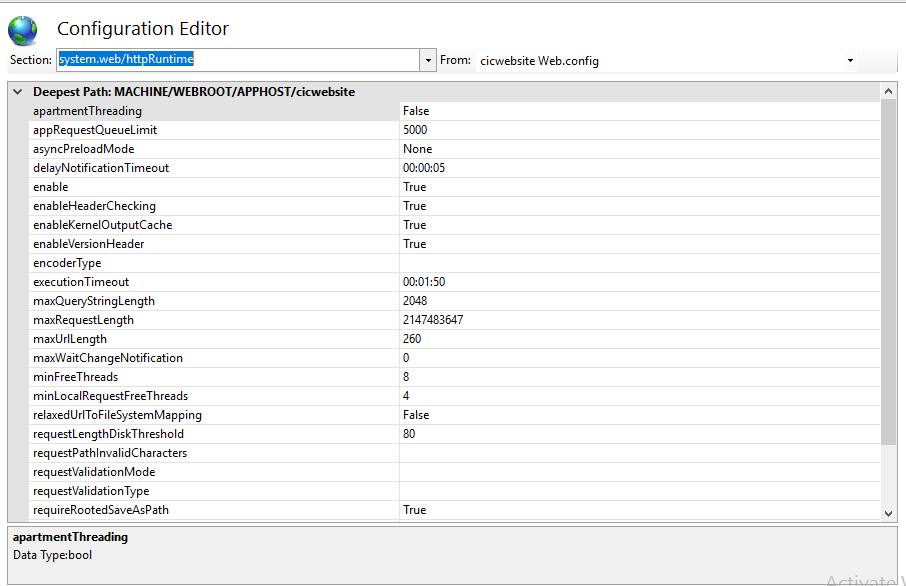
* Bước 1: Cài đặt Hosting cho .Net Core (file DotNetCore.2.0.9-WindowsHosting.exe và file dotnet-runtime-3.1.4-win-x64.exe).
* Bước 2: Cài
* Bước 3: Tạo thư mục API trên server (C:\\Website).
* Bước 4: Copy toàn bộ source code vào thư mục
* Bước 5: Điều chỉnh lại cấu hình trong file appsetting.json kết nối tới database live.
  + Mở url trong browser: <http://10.96.60.143/extra/CodecHelper>
  + Đăng nhập với username và mật khẩu của gdv.01
  + Khung test: nhập vào chuỗi cấu hình connection như ví dụ: “Data Source={IP};User ID={username};Password={password};Initial Catalog={dbname};”
  + Nhập haskey (thẻ HasheKeyDBContext trong file appsetting.json)
  + Chọn encode, click convert
  + Copy kết quả trong khung result thay thế vào thẻ DefaultConnection, BackupConnection, ExtendConnection
  + Điều chỉnh value của key ClientNodeJSAuthorize và SSOURL trỏ về IP của server và port của NodeJS
* Bước 6: Điều chỉnh các key trong appsetting.json
  + fileSaveDirectory: *đường dẫn vật lý đến thư mục FILESAVE trong project*
  + jsonwriteText : *đường dẫn vật lý đến thư mục JSONPARSE trong project*
  + LogPath: *đường dẫn vật lý đến thư mục log trong project*
  + HostName: *domain API live của BCTC*
  + ClientNodeJSSocket: *IP SSO-Server live*
  + ClientNodeJSSocketPort: *Port SSO-Server live (port 443)*
  + SSOURL: https://*[IP SSO-Server live]*/authorizeUser
  + ClientNodeJSAuthorize: https://*[IP SSO-Server live]*
* Bước 7: Tạo IIS Cho API
* Bước 8: Cấu hình application pool của IIS cho API mục .NET CLR version là No Managed Code
* Bước 9: Chọn vào website trên IIS và chọn Configuration Editor



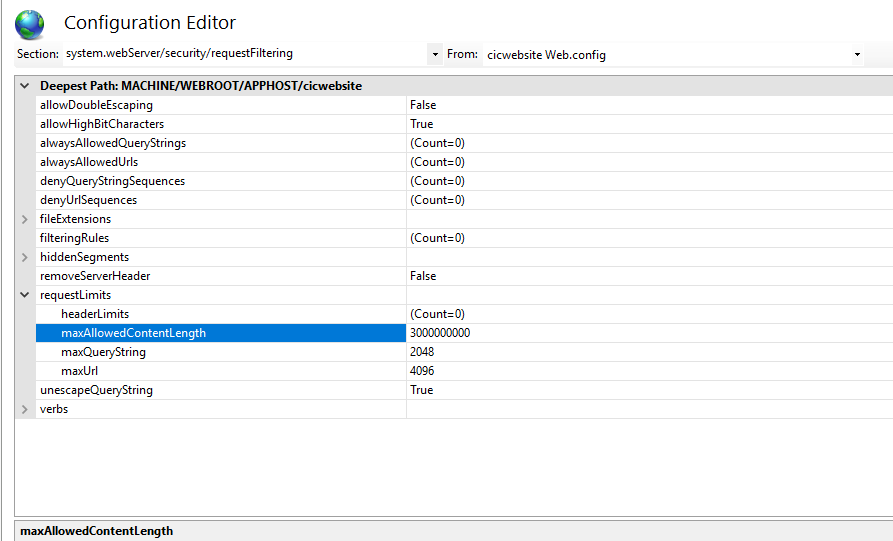
* Bước 10: Chọn Section system.webServer => serverRuntime
* Bước 11: Chỉnh 2 tham số sau và click Apply
  + uploadReadAheadSize: 2147483647
  + maxRequestEntityAllowed: 2147483647



* Bước 12: Chọn Section system.web => httpRuntime
* Bước 13: Chỉnh tham số sau và click Apply
  + maxRequestLength: 2147483647

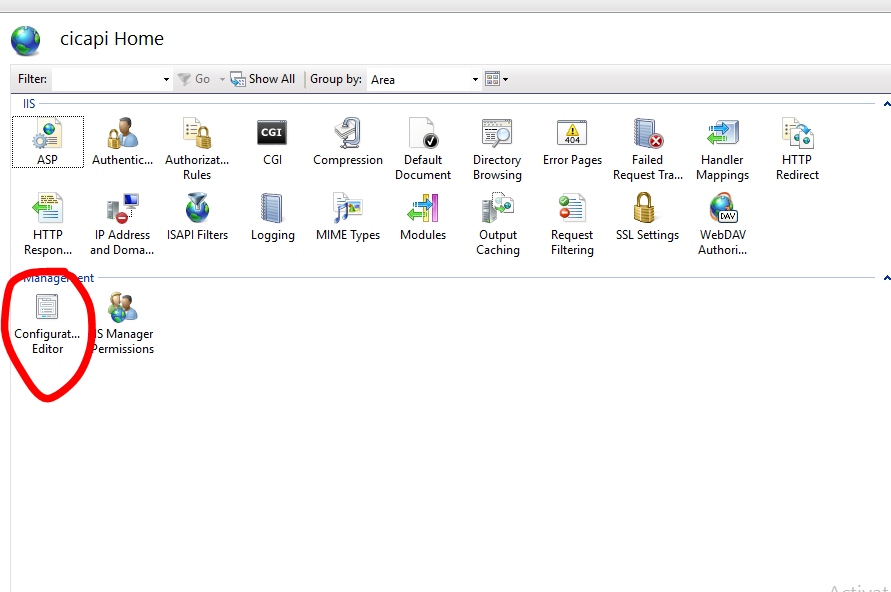


* Bước 14: Chọn Section system.webServer => security => requestFiltering
* Bước 15: Click vào mũi tên mục requestLimits
* Bước 16: Chỉnh tham số sau và click Apply
  + maxAllowContentLength: 3000000000

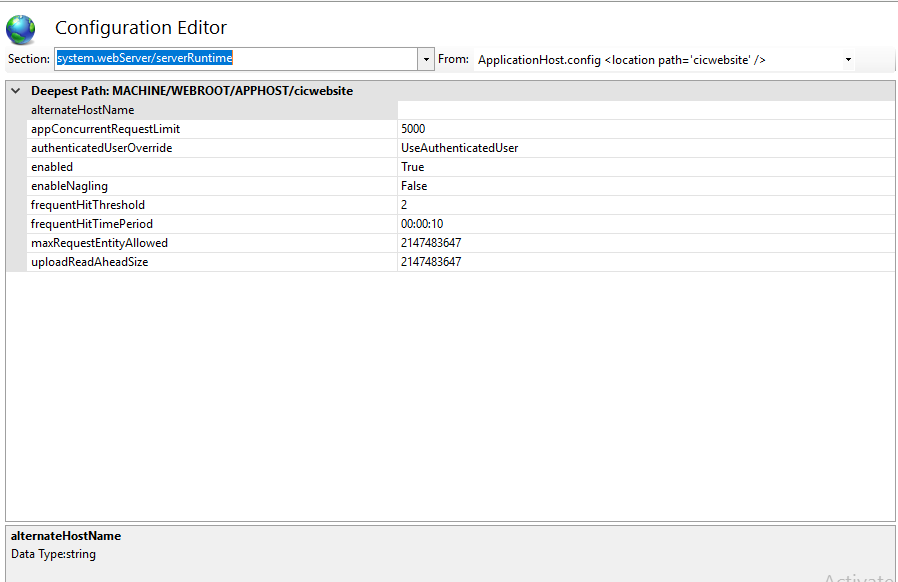


1. Cài đặt Web BCTC

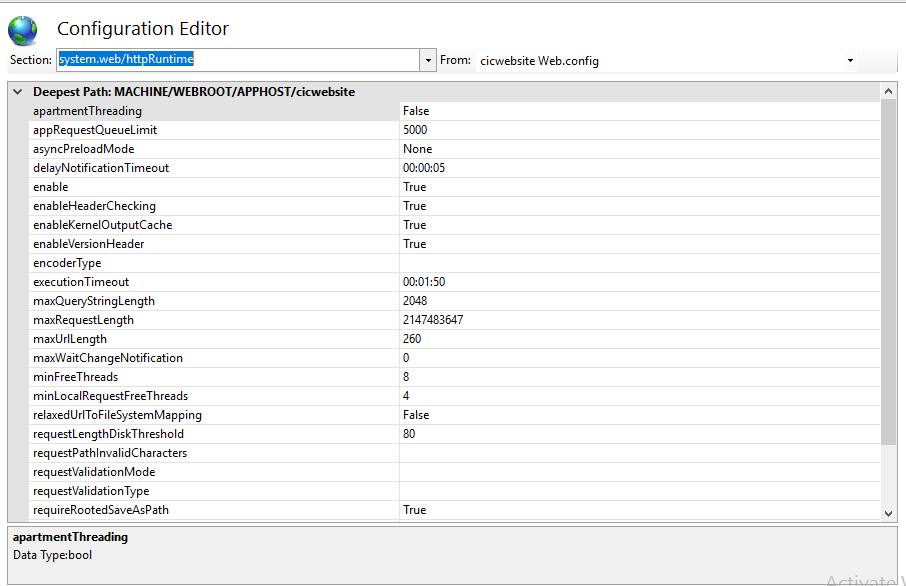
* Bước 1: Tạo thư mục Web trên server server (C:\\Website\).
* Bước 2: Copy toàn bộ source code vào thư mục
* Bước 3: Tạo IIS Cho Web
* Bước 4: Chỉnh appsettings.json trỏ đường dẫn Service API BCTC và SSO Server đến API và Server SSO.
  + ClientUrlAPI: IP đến server API BCTC
* Bước 5: Tạo IIS cho API. Cấu hình application pool của IIS cho API mục .NET CLR version là No Managed Code
* Bước 6: Chọn vào website trên IIS và chọn Configuration Editor



* Bước 7: Chọn Section system.webServer => serverRuntime
* Bước 8: Chỉnh 2 tham số sau và click Apply
  + uploadReadAheadSize: 2147483647
  + maxRequestEntityAllowed: 2147483647



* Bước 9: Chọn Section system.web => httpRuntime
* Bước 10: Chỉnh tham số sau và click Apply
  + maxRequestLength: 2147483647



* Bước 11: Chọn Section system.webServer => security => requestFiltering
* Bước 12: Click vào mũi tên mục requestLimits
* Bước 13: Chỉnh tham số sau và click Apply
  + maxAllowContentLength: 3000000000

