

Tài liệu tích hợp EmailBuilder – aiMarketing

Version: 1.0

**Mục Lục**

[​ Lịch sử thay đổi 2](#__RefHeading___Toc1425_4179576643)

[​ I. Config chung 3](#__RefHeading___Toc1427_4179576643)

[​ II. API Kết nối aiMKT 5](#__RefHeading___Toc1429_4179576643)

[​ III. API Lấy JWT Tokens 6](#__RefHeading___Toc1431_4179576643)

[​ IV. API Hủy kết nối 7](#__RefHeading___Toc1433_4179576643)

[​ V. API lấy license gửi mail 8](#__RefHeading___Toc984_3650496480)

[​ VI. API Lấy danh sách email 9](#__RefHeading___Toc1435_4179576643)

[​ VII. API Xóa email 10](#__RefHeading___Toc1437_4179576643)

[​ VIII. Gửi email 11](#__RefHeading___Toc1439_4179576643)

[​ IX. Lấy danh sách địa chỉ gửi 13](#__RefHeading___Toc1441_4179576643)

[​ X. Xóa địa chỉ người gửi 14](#__RefHeading___Toc1443_4179576643)

[​ XI. Verify email địa chỉ người gửi 15](#__RefHeading___Toc1445_4179576643)

[​ XII. Một số API ứng dụng tích hợp cần lưu ý 16](#__RefHeading___Toc2730_363930873)

[1. API lấy danh sách thuộc tính 16](#__RefHeading___Toc2732_363930873)

[2. API call back lúc lưu email 16](#__RefHeading___Toc2734_363930873)

[3. Sửa email 16](#__RefHeading___Toc2736_363930873)

[4. Tạo mới email 16](#__RefHeading___Toc2738_363930873)

[5. Truyền thêm custom-args 16](#__RefHeading___Toc2847_363930873)

# Lịch sử thay đổi

|  |  |  |
| --- | --- | --- |
| Ngày | Người sửa | Nội dung sửa |
| 02/12/2022 | Bùi Văn Minh | Khởi tạo dữ liệu |
| 06/12/2022 | Vương Ngọc Quân | Thêm API kết nối |
| 02/12/2022 | Bùi Văn Minh | Bổ sung api lấy license và api tạo, sửa email, thống nhất response trả về |
| 28/12/2022 | Bùi Văn Minh | Sửa api gửi email: Bổ sung CustomArgs cho ứng dụng tích hợp có thể custom param cần call back khi nhận event |
| 06/01/2023 | Bùi Văn Minh | Cập nhật URL |
|  |  |  |
|  |  |  |

# **I. Config chung**

* Header gồm:
  + *Content-Type: application/json*
  + *x-clientid: được aiMKT cấp cho ứng dụng tích hợp*
  + *x-clientsecret: được aiMKT cấp cho ứng dụng tích hợp*
  + *x-connectionkey: ứng dụng lấy được khi thực hiện kết nối*

- URL các môi trường

|  |  |  |  |
| --- | --- | --- | --- |
| Key | TestLocal | TestOnline | Product |
| HOST\_OPEN\_API | [http://apps.amispdc.misa.local/aimkt/APIS/OpenAPI](https://apps.amispdc.misa.local/aimkt/APIS/OpenAPI) | **https://testamisapp.misa.vn/aimkt/APIS/OpenAPI** | http://aimopenapi.misa.vn.local |
| HOST\_PARTNER\_API | [http://apps.amispdc.misa.local/aimkt/APIS/MailDuckPartnerAPI](https://apps.amispdc.misa.local/aimkt/APIS/MailDuckPartnerAPI) | **https://testamisapp.misa.vn/aimkt/APIS/MailDuckPartnerAPI** | https://mailduck.ladizone.com/partner |
| Kafka | "bootstrapservers": "kafka-headless.aimarketing:9092",  “Topic”: “TopicSendEmailBuilder”  ----------------------- Topic nhận event: “MailDuckToCRM” | "bootstrapservers": "10.16.186.17:9092" ,  “Topic”: “TopicSendEmailBuilder” ----------------------- Topic nhận event: “MailDuckToCRM” | "bootstrapservers": "172.16.183.139:9092,172.16.183.140:9092,172.16.183.141:9092" ,  “Topic”: “TopicSendEmailBuilder” ----------------------- Topic nhận event: “MailDuckToCRM” |
| UI soạn thảo email | [https://apps.amispdc.misa.local/email-builder/email](https://apps.amispdc.misa.local/email-builder/email/template?authorization) | https://testamisapp.misa.vn/email-builder/email | https://amisapp.misa.vn/email-builder/email |

- Filter trong các API lọc:

+ Các toán tử gồm:

const OPERATOR = {

**HAS\_VALUE**: 'hasvalue', **IS\_NULL\_OR\_EMPTY**: 'isnullorempty',

**CONTAINS**: 'contains', **NOT\_CONTAINS**: 'notcontains',

**STARTS\_WITH**: 'startswith', **ENDS\_WITH**: 'endswith',

**IS\_NULL**: 'isnull', **NOT\_NULL**: 'notnull',

**IN**: 'in', **NOT\_IN**: 'notin',

**NOT**: '<>', **EQUAL**: '=',

**NOT\_EQUAL**: '!=', **GREATER**: '>',

**AND**: 'AND', **OR**: 'OR',  
};

+ Một điều kiện lọc gồm 3 tham số: Tên key, toán tử, giá trị (tùy toán tử mà có cần hay không). Tên key trùng với tên field của entity:

vd: ['EmailStatus', =, EmailStatus.Draft]

Để lọc nhiều điều kiện thì điều kiện lọc như sau:

filter =

[

[Điều kiện 1],

**OR**,

[

[Điều kiện 2],

**AND**,

[Điều kiện 3],

...

],

...

]

=> Trong body request cần: btoa(JSON.stringify([filter]))

- Response trả về chung gồm:   
{  
 “IsSuccess”: true || false  
 “Data”:   
 {

...

}  
}

# **II. API Kết nối aiMKT**

|  |  |
| --- | --- |
| API | [**POST**] [**HOST\_OPEN\_API**]/v3/clients |
| Header | Dùng config chung (ngoại trừ x-connectionkey không cần truyền) |
| Body | {  CompanyID: GUID, // ID công ty CompanyCode: string, // Mã công ty CompanyName: string, // Tên công ty  ConnectionKey: string // OPTIONAL: key cho phép aiMKT kết nối ngược lại ứng dụng nếu có  } |
| Response | - Thành công, StatusCode = 200:  {  IsSuccess: true,  Code: 200,  Data: string // là x-connectionkey dùng để gọi tới các API khác  }  - Thất bại, StatusCode = 400 {  IsSuccess: false,  ErrorType: 99,  Code: 400,  } |
| Chú ý |  |

# **III. API Lấy JWT Tokens**

|  |  |
| --- | --- |
| API | [**GET**] [**HOST\_PARTNER\_API**]/v3/clients/token |
| Header | Dùng config chung |
| Body |  |
| Response | - Thành công  {  IsSuccess: true,  Code: 200,  Data: string // là chuỗi JWT có hiệu lực 1 ngày  }  - Thất bại {  IsSuccess: false,  ErrorType: 99,  Code: 400  } |
| Chú ý | Chuỗi JWT sẽ được bên tích hợp sử dụng để điều hướng sang trang UI của EmailBuilder, bằng cách thêm vào tham số đường dẫn:  ?authorization={JWT} |

# **IV. API Hủy kết nối**

|  |  |
| --- | --- |
| API | [**DELETE**] [**HOST\_OPEN\_API**]/v3/clients |
| Header | Dùng config chung |
| Body |  |
| Response | * Thành công {   IsSuccess: true,  Code: 200,  }   * Thất bại {   IsSuccess: false,  ErrorType: 99,  Code: 400,  } |
| Chú ý |  |

# V. API lấy license gửi mail

|  |  |
| --- | --- |
| API | [**GET**] [**HOST\_OPEN\_API**]/api/email/quota/{tenantID} |
| Header | Dùng config chung |
| Body |  |
| Response | * Thành công, vd:  {   IsSuccess: true,  Code: 200,  Data:   {  "LicenseType": ..., // Loại thuê bao: 0 -> Dùng thử, 1 -> trả phí có giới hạn, 2 -> unlimit  "CountQuota": ..., //Số lượng quota gói license  "CountQuotaUsed" : ..., //Số lượng quota đã sử dụng  "CountQuotaActive": ..., //Số lượng quota còn lại  "StartDate": ..., //Thời gian bắt đầu => vd: "08/09/2022 00:00:00"  "EndDate": ..., //Thời gian kết thúc  “IsLicense”: true || false // true => còn hạn, false => hết hạn  } }   * Thất bại, StatusCode = 400: {   IsSuccess: false,  ErrorType: 99,  Code: 400,  } |
| Chú ý | - Nếu trả về là -1 thì license đó không giới hạn vd: {  "LicenseType": ..., // Loại thuê bao: 0 -> Dùng thử, 1 -> trả phí có giới hạn, 2 -> unlimit "CountQuota": -1,  "CountQuotaUsed" : -1,  "CountQuotaActive": -1,  "StartDate": "", // không có t/g bắt đầu  "EndDate": "", // không có thời gian kết thúc  “IsLicense”: true || false // true => còn hạn, false => hết hạn } |

# **VI. API Lấy danh sách email**

|  |  |
| --- | --- |
| API | [**POST**] [**HOST\_PARTNER\_API**]/api/v3/emails/filter |
| Header | Dùng config chung |
| Body | {  PageIndex: 1,  PageSize: 10,  Filter: btoa(JSON.stringify(['EmailStatus', =, EmailStatus.Draft])), Sort: '+EmailName',  } |
| Response | - Thành công, StatusCode = 200:  {  IsSuccess: true,  Code: 200,  Data: {  PageData: [{EmailID, EmailName, EmailStatus, EmailThumbnail, ModifiedDate},…{}],  Total: …  } }   * Thất bại, StatusCode = 400: {   IsSuccess: false,  ErrorType: 99,  Code: 400,  } |
| Chú ý | - Trạng thái email gồm: Email nháp (Draft) = 0 Email đã gửi (Sented) = 1  Email đang gửi (Processed) = 4 Email gửi thất bại (Fail) = 5 |

# **VII. API Xóa email**

|  |  |
| --- | --- |
| API | [**DELETE**] [**HOST\_PARTNER\_API**]/api/v3/emails/{emailID} |
| Header | Dùng config chung |
| Body |  |
| Response | * Thành công,   {  IsSuccess: true,  Code: 200,  Data: emailID }   * Thất bại: {   IsSuccess: false,  ErrorType: 99,  Code: 400 || 404,  } |

# **VIII. Gửi email**

|  |  |
| --- | --- |
| API | [**POST**] [**HOST\_PARTNER\_API**]/api/v3/emails/send |
| Header | Dùng config chung |
| Body | {  EmailID**\***: guid, // ID email cần gửi EmailFromAddress**\***: string, // địa chỉ người gửi EmailFromName**\***: string, // tên người gửi  EmailReplyAddress: string?, // địa chỉ nhận email phản hồi của khách  TenantID: guid?,  Contacts: [ // danh sách liên hệ muốn gửi, danh sách < 500/request  {  ContactID: guid?,  Email**\***: string, // địa chỉ khách hàng nhận email  DynamicColumn: Dictionary<string, object>, // trường trộn  CustomArgs: Dictionary<string, object> // trường cần call back  },…{} ]  } |
| Response | * Thành công, StatusCode = 200, data trả về là emailID được gửi   {  IsSuccess: true,  Code: 200,  Data: emailID }   * Thất bại: {   IsSuccess: false,  ErrorType: 99, // 7, 19, 34, 17, 8, 20, 10, 78, 6, 99  Code: 400,  }  StatusCode 400, data trả về là mã lỗi tương ứng, mã lỗi bao gồm: 7, 19, 34, 17, 8, 20, 10, 78, 6, 99 |
| Chú ý | * Các field có \* là trường bắt buộc * Mã lỗi:   7: Không tìm thấy email để gửi,  19: Thiếu chủ để email gửi,  34: EmailFromAddress chưa được verify, 17: EmailFromName trống,  8: Email đã bị xóa,  20: Email đang gửi,  10: Không có người nhận,  78: Danh sách liên hệ cần gửi vượt quá số lượng cho phép (<=500),  6: Gửi email lỗi,  99: Gặp exception |

# **IX. Lấy danh sách địa chỉ gửi**

|  |  |
| --- | --- |
| API | [**POST**] [**HOST\_OPEN\_API**]/api/email-sender-address/filter |
| Header | Dùng config chung |
| Body | {  PageIndex: 1,  PageSize: 10,  Filter: btoa(JSON.stringify(['Status', =, ‘EmailVerifyStatus.Verify’])), Sort: ‘-ModifiedDate’,  } |
| Response | - Thành công, StatusCode = 200:  {  IsSuccess: true,  Code: 200,  Data: {  PageData: [{**Entity** EmailSenderAddress },…{}],  Total: …  }  }   * Thất bại: {   IsSuccess: false,  ErrorType: 99,  Code: 400,  } |
| Chú ý |  |
| Entity | public class EmailSenderAddress : BaseModel  {  Guid EmailSenderAddressID; string Email; // Địa chỉ email verify Guid TenantID;  EmailVerifyStatus Status; // chưa verify = 0, đã verify = 1 string SenderName; // Tên người gửi  } |

# **X. Xóa địa chỉ người gửi**

|  |  |
| --- | --- |
| API | [**DELETE**] [**HOST\_OPEN\_API**]/api/email-sender-address |
| Header | Dùng config chung |
| Body | {  EmailAddess: “…” // địa chỉ email cần xóa  } |
| Response | * Thành công, StatusCode = 200   {  IsSuccess: true,  Code: 200,  }  Thất bại: {  IsSuccess: false,  ErrorType: 99, // 69  Code: 400,  }  ErrorType = 69 => email đang được dùng để gửi ở 1 mail nào đó ErrorType = 99 => Khi gặp exception |

# **XI. Verify email địa chỉ người gửi**

|  |  |
| --- | --- |
| API | [**POST**] [**HOST\_OPEN\_API**]/api/email-sender-address |
| Header | Dùng config chung |
| Body | {  EmailAddress: “…”, // địa chỉ cần verify SenderName: “…”, Tên người gửi  } |
| Response | * Gửi verify sang aws thành công, StatusCode = 200:   {  IsSuccess: true,  Code: 200,  }  Thất bại: {  IsSuccess: false,  ErrorType: 99, // 60,59,33  Code: 400,  }  ErrorType: 60 // email sender đã verify ErrorType: 59 // quá license  ErrorType: 33 // lưu lỗi  ErrorType: 99 // Gặp exception |
| Chú ý |  |
| Entity | public class EmailSenderAddress : BaseModel  {  Guid EmailSenderAddressID; string Email; // Địa chỉ email verify Guid TenantID;  EmailVerifyStatus Status; // chưa verify = 0, đã verify = 1 string SenderName; // Tên người gửi  } |

# XII. Một số **API** ứng dụng tích hợp cần **lưu ý**

## 1. API lấy danh sách thuộc tính

- Endpoint: ...

- Method:...  
- Body:...

- Header:...

- QueryParam:...

## 2. API call back lúc lưu email

- Endpoint: ...

- Method:...  
- Body:...

- Header:...

- QueryParam:...

## 3. Sửa email

- **Qua UI**: {Domain}/email-builder/email/editor2/{emailID}?authorization={token}

## 4. Tạo mới email

**-Qua UI:**{Domain}/email-builder/email/template?authorization={token}

## 5. Truyền thêm custom-args

Khi tạo mới email hoặc lưu email các ứng dụng tích hợp muốn email-builder call back lại thì truyền thêm vào param vd:

- Khi tạo mới email:

{Domain}/email-builder/email/template?authorization={token}&key1=value1&key2=value2....

- Khi sửa email:

{Domain}/email-builder/email/editor2/{emailID}?authorization={token}&key1=value1&key2=value2....