



API

▼ /store-integration/{customerId}/to-connect-stores

▼ GET

접속한 고객(seller)가 연동 가능한 store들의 목록을 리턴해준다.

▼ Request

- content-type : Application/Json
- query string : 없음
- body : 없음.

▼ Response

statusCode : 200

```
{
  "data": [
    "Amazon US",
    "Rakuten JP",
    "Shopify",
    "Amazon kr",
    "Rakuten Kr"
  ]
}
```

response

Name	Type	discription
data	List String	seller가 미리 axB와 연동하기로한 스토어들의 리스트.

▼ /store-integration/{customerId}/connected-stores

▼ GET

접속한 고객(seller)이 연동한 store들의 목록을 리턴해준다.

▼ Request

- content-type : Application/Json
- query string : 없음.
- body : 없음.

▼ Response

statusCode : 200

```
{
  "message": "Successfully get your Connected Stores",
  "data": "data": [
    {
      "storeName": "Amazon_us",
      "keys": {
        "refreshToken": "Atzr|IwEBIBKTFWCYza0JHwM0TDXjwUAKdETvbuP_oweKHZ1qRP3Z0E6PLQ4RoZEh5H4GgcNSHY4i1K1KQMvie1a",
        "expires": {
          "expiresValue": "none",
          "available": true
        }
      }
    },
    {
      "storeName": "Rakuten_africa",
      "keys": {
        "licenseKey": "SL389938_fA5G3iKwKIg8wl8R",

```

```

        "serviceSecret": "SP389938_hX0znLmmeVrz8ntF"
      },
      "expires": {
        "expiresValue": "2021-07-25",
        "available": false
      }
    },
    {
      "storeName": "Rakuten",
      "keys": {
        "licenseKey": "SL389938_fA563iKwKig8wl8R",
        "serviceSecret": "SP389938_hX0znLmmeVrz8ntF"
      },
      "expires": {
        "expiresValue": "2025-07-30",
        "available": true
      }
    }
  ]
}

```

response

<u>Name</u>	Type	discription
<u>message</u>	String	성공했다는 메시지
<u>data</u>	List	seller가 연결한 스토어들의 키 및 만료일에 관한 데이터 리스트
<u>storeName</u>	String	Sales Channel
<u>licenseKeyValue</u>	String	rakuten의 licenseKey
<u>serviceSecretValue</u>	String	rakuten의 serviceSecret
<u>licenseKeyExpires</u>	String	키의 만료일
<u>available</u>	Boolean	만료된 키인지? true: 이용가능, false: 이용 불가(만료)

▼ /store-integration/{customerId}/{storeName}/store-alias

▼ POST

스토어 별칭을 지정한다.

▼ Request

- content-type : Application/Json
- body :

```

{
  "storeAlias": "axB"
}

```

▼ Response

statusCode : 200

```

{
  "data": {
    "connectedStores": {
      "M": {
        "Rakuten_us": {
          "M": {
            "storeAlias": {
              "S": "axB"
            }
          }
        }
      }
    }
  }
}

```

▼ /store-integration/{customerId}/{storeName}/store-alias/duplicate-check

▼ POST

스토어 별칭의 중복검사를 한다.

▼ Request

- content-type : Application/Json
- body :

```
{
  "storeAlias": "axB"
}
```

▼ Response

statusCode : 200

body : x

statusCode : 400

```
{
  "message": "store alias is duplicated!"
}
```

▼ /store-integration/{customerId}/rakuten/validation-key

▼ POST

라쿠텐의 key값의 유효성 검사를 진행한다.

▼ Request

- content-type : Application/Json
- body :




```
{
  "serviceSecret": "aaa",
  "licenseKey": "bbb"
}
```

▼ Response

statusCode : 200

```
{
  "data": "ok"
}
```

response

 Name	 Type	 discription
<u>data</u>	String	key validation의 성공 여부

▼ /store-integration/{customerId}/rakuten/apikey

▼ POST

라쿠텐의 key값과 만료일을 저장한다.

▼ Request

- content-type : Application/Json
- query string : 없음.
- body :

```
{
  "store": "Rakuten",
  "licenseKey": {
    "expires": "2021-07-19",
    "value": "aaaa"
  },
  "serviceSecret": {
    "expires": "2021-07-20",
    "value": "bbbb"
  }
}
```

body

Aa Name	Type	discription
<u>licenseKey</u>	Obect	licenseKey 객체
<u>serviceSecret</u>	Obect	serviceSecret 객체
<u>expirationDate</u>	Date	lisenseKey의 만료일
<u>value</u>	String	키 값

▼ Response

statusCode : 200

```
{
  "data": {
    "connectedStores": {
      "M": {
        "Rakuten": {
          "M": {
            "apis": {
              "M": {
                "licenseKey": {
                  "M": {
                    "value": {
                      "S": "aaaa"
                    },
                    "expires": {
                      "S": "2021-07-19"
                    }
                  }
                },
                "serviceSecret": {
                  "M": {
                    "value": {
                      "S": "bbbb"
                    },
                    "expires": {
                      "S": "2021-07-20"
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}
```

response

Aa Name	Type	discription
---------	------	-------------

Aa Name	Type	discription
<u>connectedStores</u>	Object	dynamo에 저장된 모양

▼ /store-integration/{customerId}/amazon/custom-weblink

▼ GET

아마존 인증을 시작할 수 있는 weblink를 제공한다.

▼ Request

- content-type : Application/Json
- query string
 - storeName
- body : 없음

▼ Response

statusCode : 200

```
{
  "data": "https://sellercentral.amazon.com/apps/authorize/consent?application_id=amzn1.sp.solution.b0aac576-c737-48a5-"
}
```

response

Aa Name	Type	discription
<u>data</u>	String	weblink

▼ /store-integration/{customerId}/amazon/refresh-token

▼ GET

customerId에 해당하는 seller의 refresh token을 제공한다.

▼ Request

- content-type : Application/Json
- query string
 - storeName
- body : 없음

▼ Response

statusCode : 200

```
{
  "data": "Atza|IwEBIJDJwBYwh3GojRPJjDaWJoGYkcBv8U9pntUGwEGmZwIZDUKpVLIyMl5cMnh5Z8agP7ioN6Bx7BtV49fstt0jz8jQLxU5TvmnKsr"
}
```

response

Aa Name	Type	discription
<u>data</u>	String	refreshToken

▼ /store-integration/{customerId}/amazon/isConnected

▼ PUT

DynamoDB의 해당 데이터에 isConnected 속성의 값을 true로 바꿔준다.

▼ Request

- content-type : Application/Json
- query string
 - storeName

▼ Response

statusCode : 200

```
{
  "data": {
    "connectedStores": {
      "M": {
        "Amazon_us": {
          "M": {
            "isConnected": {
              "BOOL": true
            }
          }
        }
      }
    }
  }
}
```

▼ /store-integration/{customerId}/shopify/install-link

▼ GET

shopify 인증을 시작할 수 있는 custom-link를 제공한다.

▼ Request




- content-type : Application/Json
- query string
 - storeName
- body : 없음

▼ Response

statusCode : 200

```
{
  "data": "https://acrossb.myshopify.com/admin/oauth/install_custom_app?client_id=ddf31d1d9567b9280f6ecde2711a56de&sign"
}
```

response

 Name	 Type	 discription
<u>data</u>	String	install link

▼ /store-integration/{customerId}/shopify/access-token

▼ GET

customerId에 해당하는 seller의 access token과 그 유효성을 제공한다.

▼ Request

- content-type : Application/Json
- query string
 - storeName
- body : 없음

▼ Response

statusCode : 200

```
{
  "data": {
    "accessToken": "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",
    "isValidate": true
  }
}
```

response

<u>Name</u>	Type	discription
<u>accessToken</u>	String	accessToken
<u>isValidate</u>	Boolean	accessToken이 유효한지

▼ /store-integration/{customerId}/shopify/isConnected

▼ PUT

DynamoDB의 해당 데이터에 isConnected 속성의 값을 true로 바꿔준다.

▼ Request

- content-type : Application/Json
- query string
 - storeName

▼ Response

statusCode : 200

```
{
  "data": {
    "connectedStores": {
      "M": {
        "Shopify_us": {
          "M": {
            "isConnected": {
              "BOOL": true
            }
          }
        }
      }
    }
  }
}
```